National Library of Canada
Bibliothèque nationale du Canada

Canadian Theses Service
Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

NL-339 (r. 88/04) c

Canada

UNIVERSITY OF ALBERTA

Identification and Control of Multivariable Systems

BY

Viral Mahendra Maniar

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN

PROCESS CONTROL

DEPARTMENT OF CHEMICAL ENGINEERING

EDMONTON, ALBERTA

SPRING, 1992

National Library of Canada
Bibliothèque nationale du Canada

Canadian Theses Service
Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

ISBN 0-315-73246-6

Canada

UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR:	Viral Mahendra Maniar
TITLE OF THESIS:	Identification and Control of Multivariable Systems
DEGREE:	Master of Science
YEAR THIS DEGREE GRANTED:	1992

Permission is hereby granted to the University of Alberta Library to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves all other publication and other rights in association with the copyright in the thesis, and except as hereinbefore provided neither the thesis nor any substantial portion thereof may be printed or otherwise reproduced in any material from whatever without the author's prior written permission.

(SIGNED) Viral

2B, Chandravilla Apartments

Gulmohur Cross Lane # 5

Juhu Scheme, Vile-Parle (west)

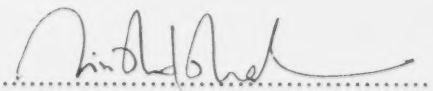
Bombay 400 049, INDIA.

DATED: 19.

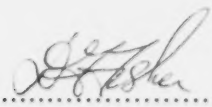
UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

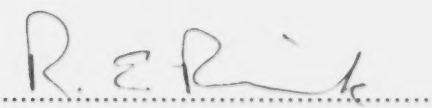
The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled **Identification and Control of Multivariable Systems** submitted by **Viral Mahendra Maniar** in partial fulfillment of the requirements for the degree of **Master of Science** in **Process Control**.



Prof. S. L. Shah (co-supervisor)



Prof. D. G. Fisher (co-supervisor)



Prof. R. E. Rink

DATE: April 24, 1992

To

my Guru Param Pujya Bramhaswaroop Shri Pramukh Swami Maharaj

and

my Mummy and Pappa

ABSTRACT

This thesis deals with on-line identification and constrained long range predictive control of multivariable systems. It extends a recently proposed augmented upper-diagonal factorization identification (AUDI) algorithm of Niu *et al.* (1990), to identify input-output models of multivariable systems with distinct time-delays. The multi-input multi-output (MIMO AUDI) algorithm can identify the process model order and process parameters simultaneously. The MIMO AUDI algorithm is built on the idea of decomposing a MIMO system into as many multi-input single output (MISO) subsystems as the number of outputs and then identifying each MISO subsystem separately. This strategy allows large identification problems to be decomposed into a series of smaller identification problems and the distinct time-delays in the system to be handled elegantly.

One of the important fundamental issues in the identification of multivariable systems is the representation of the process. This thesis discusses various canonical forms for representing multivariable systems. A transformation is developed by which MIMO systems represented in transfer function matrix (TFM) form can be transformed into an auto-regressive moving average (ARMA) form. These results then lead to the decomposition of an n x m MIMO system into n MISO subsystems. The latter configuration allows one to solve a smaller-dimensional least squares optimization problem.

The performance of the new MIMO AUDI algorithm is demonstrated by evaluation using the input-output data from a real process. The extension of this algorithm by incorporating a variable forgetting factor with a lower bound on its value

has also been studied. A comparative study of identification of two different types of models, namely an auto-regressive moving average model with an exogenous input (ARMAX) versus an auto-regressive integrated moving average model with an exogenous input (ARIMAX) is discussed.

The MIMO AUDI algorithm was used on-line in closed loop with a MIMO constrained GPC algorithm to experimentally evaluate MIMO adaptive constrained GPC control of a continuous stirred tank heater. The role of various controller tuning parameters on the performance of the closed-loop system, with and without rate and amplitude constraints on the control inputs and on the outputs of the process, is also successfully illustrated. The servo and regulatory capabilities of the algorithm in controlling a marginally open-loop unstable process were studied and found to be excellent in comparison with classical multiloop PID control.

Acknowledgements

I would like to sincerely thank Dr. Sirish L. Shah for his supervision, guidance and for helping me bring my thesis in the present form. I deeply feel that without his help and motivation, I would never have been able to work in this area. I would like to thank Dr. D. Grant. Fisher for his guidance and direction. I would also like to take this opportunity to thank my very good friend and colleague Rajendra Mutha for his help and support to get me started in a area which was totally new to me.

I would also like to thank Shaohua Niu for his help with the AUD, Kunyu Kwok and Eric Lau for their help with MULTICON, Walter Boddez in helping me with any problem with the continuous stirred tank heater.

I would like to thank my friends Fengxi Zhou, Roman Walter and Mary Bourke for making my work at office very enjoyable and interesting. I would like to thank my control group for very intriguing discussions on control and in particular to Dr. Lixin Peng. I would like to thank my friends Natarajan Iyer, Sreekant Lalgudi, Ravindra Gudi, Pranob Banerjee (Dada) and Avinash Shanbhag for bearing with me. I would like to thank my best friends Manjunatha Bhat and Narendra Ravi for their constant moral support away from from home.

Financial support from the Department of Chemical Engineering and NSERC is gratefully acknowledged.

And above all I would like to express my deepest sense of gratitude to my parents because without their love and sacrifices for me over so many years, I would never have achieved anything.

Contents

1 Introduction.....1

Objective of this study.....5

Structure of thesis.....7

2 Structural representation of Multivariable systems with distinct time-delays.....10

2.1 Canonical structures of multivariable processes.....11

2.1.1 The P-canonical structure.....12

2.1.2 The V-canonical structure.....13

2.1.3 P+V canonical structure.....14

2.1.4 Observable canonical state-space representation.....18

2.1.5 Deterministic Auto-regressive Moving Average (DARMA)representation.....24

2.2 Parametric equivalence between the TFM form and the DARMA form.....25

2.2.1 An Illustrative example.....32

3 Identification of Multivariable systems.....39

3.1 Recursive Identification techniques.....40

3.1.1 Models for Identification.....41

3.1.2 Preliminaries for recursive identification.....43
3.1.3 Recursive Least Squares (RLS) technique.....44
3.1.3.1 RLS with exponential data weighting.....47
3.1.4 UDU^T factorization technique.....48
3.1.5 Augmented UD (AUD) factorization technique.....52
3.2 AUD for Identification of Multivariable systems (MIMO AUDI)...57
3.2.1 Open-loop study of MIMO AUDI.....64
3.2.1.1 Identification of the model order of theprocess.....67
3.2.1.2 Identification of ARMAX and ARIMAX models parameters with no forgetting of past data.....70
3.2.1.3 MIMO AUDI with constant forgetting factor.....80
3.2.1.4 MIMO AUDI with variable forgetting factor.....84

4 Generalized Predictive Control of Multivariable systems.....93
4.1 The Multivariable GPC (MGPC) algorithm.....94
4.1.1 The CARIMA process model.....94
4.1.2 Long-range prediction in MGPC.....95
4.1.3 The predictive control law of MGPC.....96
4.1.4 MGPC algorithm with constraints.....100
4.1.5 Outputweightinginmgpc.....102

5 Experimentalevaluationof MultivariableAdaptiveGPC.....104
5.1 Implementation details.....105
5.1.1 Hardware details.....105
5.1.2 The QNX operating system.....106
5.1.2.1 The structure of QNX.....106
5.1.3 The MULTICON.....108
5.1.3.1 The structure of MULTICON.....109
5.2 Adaptive control of continuous stirred tank heater using MAdGPC algorithm.....111

5.2.1 Process equipment description....111
5.2.2 Implementation of MIMO Adaptive GPC....113
5.2.3 Experimental runs on CSTH....115
5.2.3.1 Unconstrained MAdGPC with variable forgetting factor....115
5.2.3.2 Effect of lower bound on variable forgetting factor....119
5.2.3.3 Effect of maximum prediction horizon (N_2)....124
5.2.3.4 Effect of control input weighting....126
5.2.3.5 Effect of output weighting....130
5.2.3.6 Effect of T-filter....132
5.2.3.7 Effect of input rate constraints....134
5.2.3.8 Effect of input amplitude constraints....136
5.2.3.9 Analytically solved input amplitude and rate constraints on MAdGPC....138
5.2.3.10 Effect of output constraints on MAdGPC....142
5.2.3.11 The Servo and Regulatory properties of Constrained MAdGPC....145
5.2.3.12 Multi-loop PID controller on a 2 x 2 process....149

6 Conclusions....152
6.1 Recommendations for future work....156

References....157

List of Figures

1.1 Schematic representation of a typical adaptive control system....3

1.2 Setpoint, output and control trajectories in GPC....5

2.1 A general multivariable process with n outputs and m inputs....11

2.2(a) P-canonical structure of a 2 x 2 system....12

2.2(b) V-canonical structure of a 2 x 2 system....12

2.3 P+V canonical structure of a 2 x 2 system....15

3.1 A diagram to show the optimal model order of a system....63

3.2 Output data from the continuous stirred tank heater....68

3.3 Input data from the continuous stirred tank heater....68

3.4 Loss functions for MISO subsystems corresponding to temperature and level loops respectively of the continuous stirred tank heater....69

3.5 Actual and estimated temperature from ARMAX model identification without forgetting factor....72

3.6 Actual and estimated level from ARMAX model identification without forgetting factor....72

3.7 Parameter estimates of the MISO subsystem corresponding to the temperature loop in ARMAX representation without forgetting factor....73

3.8 Parameter estimates of the MISO subsystem corresponding to the level loop in ARMAX representation without forgetting factor....73

3.9 Estimates of the error of closure in temperature and level in ARMAX representation without forgetting factor....74

3.10 Trace of the covariance matrix of the data vector in both MISO subsystems in ARMAX representation without forgetting factor....74

3.11 Actual and estimated temperature and level from ARIMAX model identification without forgetting factor....77

3.12 Parameter estimates of MISO subsystems corresponding to the temperature and the level loops in ARIMAX representation without forgetting factor 78
3.13 Trace of the covariance matrix of the data vector in both MISO subsystems in ARIMAX representation without forgetting factor.....79
3.14 Actual and estimated temperature and level from ARMAX model identification with a constant forgetting factor of 0.95.....81
3.15 Parameter estimates of MISO subsystems corresponding to the temperature and level loops in ARMAX representation with a constant forgetting factor of 0.95...82
3.16 Estimates of the error of closure in temperature and level in ARMAX representation with a constant forgetting factor of 0.95.....83
3.17 Trace of the covariance matrix of the data vector in both MISO subsystems in ARMAX representation with a constant forgetting factor of 0.95.....83
3.18 Actual and estimated temperature and level from ARMAX model identification with a variable forgetting factor 85
3.19 Parameter estimates of MISO subsystems corresponding to the temperature and level loops in ARMAX representation with a variable forgetting factor.....86
3.20 Estimates of the error of closure in temperature and level in ARMAX representation with a variable forgetting factor.....87
3.21 Trace of the covariance matrix of the data vector in both MISO subsystems in ARMAX representation with a variable forgetting factor.....87
3.22 Variation of the forgetting factor in ARMAX model identification.....88
3.23 Actual and estimated temperature and level from ARMAX model identification using a variable forgetting factor with a lower bound at 0.75.....89
3.24 Parameter estimates of MISO subsystems corresponding to the temperature and level loops in ARMAX representation using a variable forgetting factor with a lower bound at 0.75.....90
3.25 Estimates of the error of closure in temperature and level in ARMAX representation using a variable forgetting factor with a lower bound at 0.75.....91
3.26 Trace of the covariance matrix of the data vector in both MISO subsystems in ARMAX representation using a variable forgetting factor with a lower bound at 0.75.....91

3.27 Variation of the forgetting factor in ARMAX model identification.....92

5.1 Schematic representation of the continuous stirred tank heater for MIMO configuration.....112
5.2 Schematic representation of the MIMO adaptive GPC controller.....113
5.3 Schematic representation of the implementation of the MIMO adaptive GPC on the continuous stirred tank heater.....114
5.4 Unconstrained MIMO adaptive GPC with variable forgetting factor including the initializations of the adaptive controller.....116
5.5 Parameter trajectories of the MISO subsystem corresponding to the temperature loop in unconstrained MIMO adaptive GPC.....117
5.6 Parameter trajectories of the MISO subsystem corresponding to the level loop in unconstrained MIMO adaptive GPC.....117
5.7 Estimates of the error of closure in temperature and level in ARMAX representation in unconstrained MIMO adaptive GPC.....118
5.8 Effect of a lower bound on variable forgetting factor and the sigma parameter on unconstrained MIMO adaptive GPC.....120
5.9 Parameter trajectories of the MISO subsystem corresponding to the temperature and level loops due to the variation of the lower bound on forgetting factor value and the sigma parameter.....121
5.10 Estimates of the error of closure in temperature and level in ARMAX representations due to the variation of the lower bound on forgetting factor value and the sigma parameter.....122
5.11 Variation in the forgetting factor value due to a lower bound on its value and the sigma parameter.....122
5.12 Effect of maximum prediction horizon (N_2) on the performance of unconstrained MIMO adaptive GPC.....125
5.13 Effect of input weighting (Q) on the performance if unconstrained MIMO adaptive GPC.....127
5.14 Parameter trajectories of the MISO subsystems corresponding to the temperature and level loops due to variation of input weighting.....128

5.15 Estimates of the error of closure in temperature and level in ARMAX representation due to the variation of input of input weighting.....129
5.16 Effect of output weighting (y_wt) on the performance of unconstrained MIMO adaptive GPC.....131
5.17 Performance of unconstrained MIMO adaptive GPC with and without the T-filter.....133
5.18 Performance of MIMO adaptive GPC in the presence of rate constraints on inputs solved using the QPSOL optimization package 135
5.19 Performance of MIMO adaptive GPC in the presence of amplitude constraints on inputs solved using the QPSOL optimization package 137
5.20 Performance of MIMO adaptive GPC in the presence of rate constraints on inputs solved analytically.....139
5.21 Performance of MAdGPC with analytical solution of the rate constraints on inputs in the presence of disturbances.....140
5.22 Performance of MIMO adaptive GPC in the presence of amplitude constraints on inputs solved analytically.....141
5.23 Performance of MIMO adaptive GPC in the presence of amplitude constraints on the outputs solved using the QPSOL optimization package.....143
5.24 Servo and regulatory properties of MIMO adaptive constrained GPC.....146
5.25 Parameter trajectories of the MISO subsystems in ARMAX representation corresponding to the temperature and level loops in MIMO adaptive constrained GPC.....147
5.26 Estimates of the error of closures in temperature and level in ARMAX representation in MIMO adaptive constrained GPC.....148
5.27 Variation in the forgetting factor value in MIMO adaptive constrained GPC.....148
5.28 Multiloop PID implemented on the continuous stirred tank heater configured as a 2 x 2 process.....150

Chapter 1

Introduction

In the 1950's, extensive research was done on adaptive control in connection with the design of autopilots for high performance aircrafts. Unfortunately, the interest in this area diminished due to a lack of supporting theory and suitable computational facilities for implementing such algorithms in an on-line situation. Since 1970, there have been significant theoretical developments in adaptive control, with landmark contributions on self-tuning control by Astrom and Wittenmark (1973) and Clarke and Gawthrop (1975). Also, due to advances on micro-processor based systems, an abundance of inexpensive computational power has become available. These developments have renewed interest in adaptive control. Adaptive control as a result has remained an active research area over the last two decades.

Although lot of research has been done in this area, it is surprising that there is no general consensus on a formal definition of *adaptive control.* A variety of definition of "Adaptive Control" have emerged in this active area of research:

An adaptive system is any physical system that has been designed with an adaptive viewpoint (Astrom and Wittenmark, 1989).

Adaptive control is a direct aggregation of a control methodology with some form of recursive system identification (Sastry and Bodson, 1989).

An adaptive controller is one which continuously identifies (on-line) the parameters of the process as they change, and retunes the controller appropriately (Luyben, 1989).

An adaptive control system is one that automatically adjusts the controller settings to accommodate changes in the process to be controlled or its environment (Seborg *et al.*,1986).

Chemical processes are in many respects ideal candidates for adaptive control applications, since the dynamic behaviour of many common processes does vary with respect to time which inturn requires frequent retuning of the controller. For example, a catalytic process can change its characteristics due to changes in the catalyst behaviour (Lee and Weekman, 1976). There can be various other causes of changing process conditions such as, heat exchanger fouling, large and frequent disturbances as in feed composition etc., ambient variations due to rain storms or sudden drop in temperature, etc.

Also, a typical industrial chemical process is inherently non-linear in nature and yet most of the control literature is concerned with controller design for linear systems. Non-linear systems are often linearized around a fixed operating condition or a steady state and with changes in the operating conditions, the linear constant coefficient will change. Thus, the controller needs to be retuned every time the process undergoes a

change and hence an adaptive controller is justified. A typical adaptive controller as implemented on a process is schematically represented in figure 1.1.

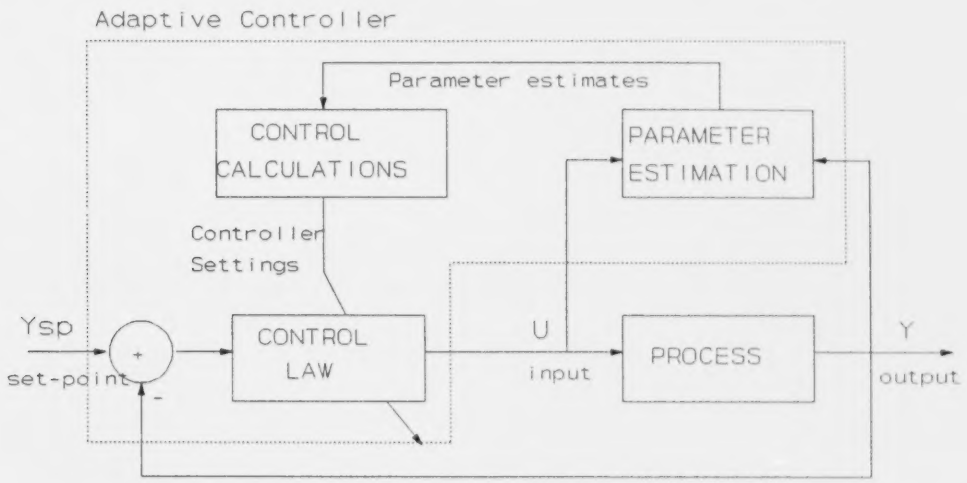


Figure 1.1 Schematic representation of a typical adaptive control system

The main components of an adaptive controller are the parameter estimation or system identification block and the control block. One of the major benefits of adaptive controllers can be the potential increase in the speed of controller commissioning. Furthermore, the use of adaptive controller can result in 'expert tuning by non-expert personnel', i.e. widespread use of adaptive controllers can increase the overall quality of control throughout the industry by making the quality of control in individual loops less dependent upon the expertise of control engineers.

Model based Predictive Control (MPC) has attracted the attention of a lot of researchers in the past decade. Richalet *et al.*(1978) described successful applications of "Model Predictive Heuristic Control". Cutler and Ramaker of Shell, designed Dynamic Matrix

Control (DMC, 1980) and reported some applications to a fluid catalytic cracker. In both these algorithms an explicit dynamic model of the plant is used to calculate future predictions of outputs from the future control actions, and hence the name *Model Predictive Control.* There are many MPC techniques which were developed for adaptive control, like Peterka's optimal predictive controller (Peterka, 1984), Ydstie's Extended Horizon Adaptive Controller (EHAC, 1984) and Extended Prediction Self-Adaptive Control (EPSAC) of DeKeyser and van Cauwenberghe (1985).

A new algorithm in the MPC category called the *Generalized Predictive Controller* (GPC) proposed by Clarke *et al.*(1987) is a general purpose algorithm. GPC can handle large, unknown, variable time-delays, non-minimum phase systems and open-loop unstable plants. It is a natural extension of the generalized minimum variance (GMV) controller of Clarke and Gawthrop (1975). It can run in an adaptive or non-adaptive mode. The control philosophy in GPC involves minimization of a multistep quadratic cost function of future predicted errors to give the predictive control law. It uses an auto-regressive integrated moving average, with an exogeneous input (ARIMAX) model structure for the process model which ensures inherent integral action (Clarke *et al.*,1987). The objective of the predictive control law is to drive the future plant outputs y(t+j) "close" to the future set-point trajectory (which is known) w(t+j) as shown in figure 1.2. This is done using a receeding-horizon approach (Clarke *et al.*,1987) for which at each sample instant t:

1. The future set-point sequence w(t+j) is calculated;
2. A set of predicted outputs y(t+j/t) with corresponding predicted system errors e(t+j)

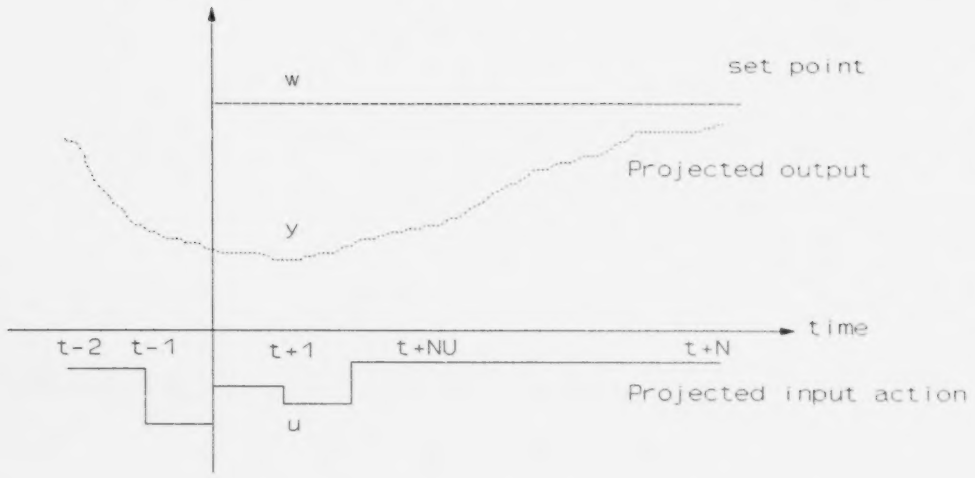


Figure 1.2 Setpoint, output and control trajectories in GPC

are computed;

3. An appropriate quadratic cost function basec on the future errors and control action is minimized, assuming that after some 'control horizon' further increments in control are zero, to provide a suggested sequence of future controls u(t+j). This idea has been borrowed from the DMC algorithm of Cutler and Ramaker (1980).

Proper selection of prediction horizon and the control horizon makes GPC relatively insensitive to unknown or variable time delays and non-minimum phase processes.

Objective of this Study

Most chemical processes are inherently multivariable in nature, that is there can be more than one output which needs to be controlled by manipulating more than one control input. On a broader base, the goal of this thesis is:

- **To study identification of multi-input multi-output (MIMO) processes**

In the identification of multivariable systems, the structural representation of a MIMO process such that it leads itself to a relatively straightforward identification is very important. This thesis develops a transformation to put the MIMO system into a form that is convenient for identification.

- **To develop a practical multivariable identification algorithm**

The augmented UD factorization (AUD) identification algorithm of Niu *et al.*(1990) is extended to MIMO systems where the time delay in each element of the transfer function matrix can be distinct. The MIMO AUD identification algorithm developed here can identify the model order of each multi-input, single output (MISO) subsystem at every time instant. This is more general than identifying just one model order for the complete MIMO system, since different MISO subsystems in a MIMO process can have different model orders depending on the relation between the outputs and the inputs.

- **To evaluate the performance of the multivariable identification algorithm on a real process**

The MIMO AUD identification algorithm based on a set of MISO model is then evaluated using input-output data obtained from a real process and the identification of different models, such as the auto-regressive moving average model with an exogenous input (ARMAX) and the auto-regressive integrated moving average model with an exogenous input (ARIMAX) model are compared. The idea of the variable forgetting

factor of Fortescue *et al.*(1981) has been adopted here to retain the alertness of the algorithm in the absence of any excitation in the process. A lower bound on the forgetting factor value is used which gives the flexibility to reduce the variance of the parameter estimates.

- **to implement MIMO adaptive GPC on a pilot-scale continuous stirred tank heater and to evaluate the performance of the combined identification and control algorithms.**

The software for constrained MIMO GPC algorithm was developed at the University of Alberta by Mutha (1990). To implement MIMO adaptive GPC algorithm, the software for MIMO AUD identification algorithm was developed in C and was implemented with MIMO GPC to work under the MULTICON task manager.

The MIMO adaptive GPC algorithm was experimentally evaluated on a continuous stirred tank heater. The effects of varying tuning parameters and the effects of constraints on this algorithm were also investigated. The performance of the algorithm for regulatory action and set-point tracking was also evaluated.

Structure of the Thesis

This thesis is mainly concerned with the development of a MIMO AUD identification algorithm and its evaluation on a real process. Another important component of this thesis is the implementation of the MIMO adaptive GPC algorithm on the continuous stirred tank heater. The thesis is organized as follows:

Chapter 2 deals with the canonical structures for representing multivariable systems. A transformation is developed to convert a MIMO system given in transfer function matrix form to an auto-regressive moving average (ARMA) model. This transformation results in a special representation of the MIMO systems, which is very convenient for identification.

Various recursive identification techniques including the recent augmented UD (AUD) identification technique of Niu *et al.*(1990) are discussed in chapter 3. The AUD algorithm is extended to identify MIMO systems with distinct time-delays labelled as the MIMO AUD algorithm for identification (MAUDI). This algorithm is modified by incorporating a variable forgetting factor to retain its alertness in the absence of excitation from the process. A lower bound on the forgetting factor value is used to keep the variance of the parameter estimates low. The various capabilities of the MIMO AUD identification algorithm such as on-line model order estimation and parameter estimation are also discussed. This algorithm is evaluated on the input-output data acquired from a real process. Identification, using the above algorithm, of different input-output models such as ARMAX and ARIMAX is also discussed.

The basic GPC algorithm is described in chapter 4 together with the derivation of the control law for the MIMO GPC (MGPC) algorithm. It also discusses the effects of constraints and output weighting in the MGPC algorithm.

Chapter 5 is devoted to the implementation and experimental evaluation of a MIMO constrained adaptive GPC algorithm. It describes the software and the hardware used including the process equipment. The effects of various tuning parameters on the

overall performance of the algorithm are discussed. The effects of constraints on the algorithm are studied and the comparative results with the unconstrained case are discussed. The set point tracking and regulatory capabilities for the unconstrained and the constrained algorithms are presented. Discussion on the choice of various tuning parameters are also given in the respective sections.

The thesis ends with chapter 6 summarizing the conclusions of the study followed by suggestions for future work.

Chapter 2

Structural representation of multivariable systems with distinct time-delays

The key elements of system identification are selection of model structure, parameter estimation and model validation (Astrom and Wittenmark, 1989). The identification of multi-input, multi-output systems is not a simple extension of the single-input, single-output case. Infact, the preliminary formulation of the multivariable identification problem in terms of selection of a suitable model structure is itself a challenging and non-trivial task. Many such parameterization issues related to model structure selection are still unresolved (Wolovich and Elliot, 1983). This chapter discusses the selection of a suitable model structure to represent multivariable systems with distinct time-delays and proposes one suitable and practically realizable structure for identification.

A multivariable system can be represented in many different but equivalent

mathematical forms. In this chapter, various commonly used input-output structures or representations of multivariable processes are discussed.

A general schematic representation of the signal flow paths in a multivariable process are shown in figure 2.1. The inputs u_i of the process may affect some or all of the outputs y_i, resulting in interactions between various loops in this process.

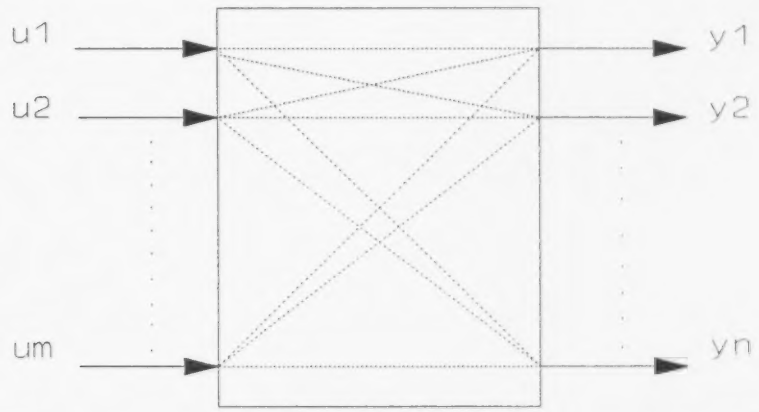


Figure 2.1 A general multivariable process with m-inputs and n-outputs

2.1 Canonical structures of multivariable processes

Two commonly used "canonical" structures which describe a multivariable process from an input-output point of view are illustrated in figure 2.2 for the specific case of a 2 x 2 system. These are known as the *P-canonical structure* and the *V-canonical structure* (Isermann R., 1981).

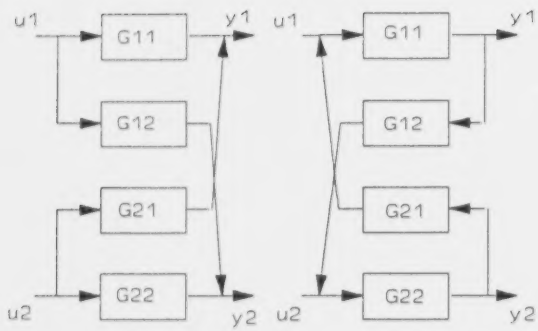


Figure 2.2 (a) P-canonical structure (b) V-canonical structure for a 2 x 2 system

2.1.1 The P - Canonical structure of a multivariable system

In the case of the P-canonical structure, each input affects every output, interactions are represented as feedforward couplings so the summation occurs at the outputs. It can also be seen that the changes in any of the transfer function elements, would influence only the corresponding output. The number of inputs can be different from the number of outputs. The P-canonical structure for the process shown in figure 2.2(a) and can be mathematically represented as:

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{2.1}$$

$$or \quad Y = G\,U \tag{2.2}$$

In this process the number of inputs is equal to the number of outputs, and hence the transfer function matrix of the process is square. If the number of inputs and outputs are unequal, the transfer function matrix becomes non-square or rectangular. It is worthwhile to note here that the transfer function elements describe only the controllable and observable parts of the process. Uncontrollable and unobservable parts of a process are not represented in the input-output or transfer functions model of the process.

2.1.2 The V - Canonical structure of the multivariable system

In the case of the V-canonical structure, each input acts directly on one output and each output acts on the remaining inputs. This structure can be defined only for systems with an equal number of inputs and outputs. Changes in one transfer function element influence the signals of all the other elements. The V-canonical structure shown in the figure 2.2(b) can be mathematically represented as:

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} G_{11} & 0 \\ 0 & G_{22} \end{bmatrix} \left\{ \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} + \begin{bmatrix} 0 & G_{21} \\ G_{12} & 0 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \right\} \tag{2.3}$$

or

$$\underline{Y} = \underline{\underline{G_H}} \left\{ \underline{U} + \underline{\underline{G_K}}\, \underline{Y} \right\} \tag{2.4}$$

where, $\underline{\underline{G}}_H$ is a diagonal matrix which contains the main elements and $\underline{\underline{G}}_K$ contains the coupling elements. Eqn 2.3 can be explicitly represented as:

$$\underline{Y} = \left\{ \underline{I} - \underline{\underline{G_H}}\,\underline{\underline{G_K}} \right\}^{-1} \underline{\underline{G_H}}\,\underline{U} \tag{2.5}$$

It can be easily shown that the transfer function matrix of the V-canonical structure is

$$\underline{\underline{G}} = \{ \underline{I} - \underline{\underline{G_H}}\,\underline{\underline{G_K}} \}^{-1} \underline{\underline{G_H}} \tag{2.6}$$

Using eqn 2.6, the V-canonical structure of a process can be converted into a P-canonical form and conversely the P-canonical structure for a square system can also be converted into a V-canonical form. The P-canonical structure is a better form of representation of a system since it is more general than the V-canonical form in its ability to represent non-square systems.

2.1.3 The P + V Canonical structure of a Multivariable system

This structure follows logically from the previous canonical structures. In P+V canonical structures, each input affects every output, and the outputs directly affect each other. There are chemical systems which need to be represented by this structure, e.g. a CSTR, where the concentration of the product and its temperature are to be controlled by manipulating the coolant flowrate and the reactant flow rate. The concentration of the product and the temperature of the product, which are the controlled outputs of the system, both directly affect each other.

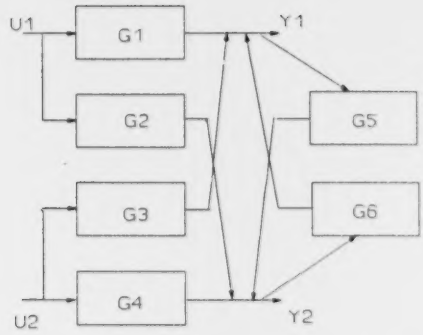


Figure 2.3 The P+V canonical structure for a 2 x 2 system

The P+V canonical structure shown in figure 2.3 can be mathematically represented as follows:

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} G_1 & G_3 \\ G_2 & G_4 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} + \begin{bmatrix} 0 & G_6 \\ G_5 & 0 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \tag{2.7}$$

or

$$\left\{ I - \begin{bmatrix} 0 & G_6 \\ G_5 & 0 \end{bmatrix} \right\} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} G_1 & G_3 \\ G_2 & G_4 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{2.8}$$

or

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & -G_6 \\ -G_5 & 1 \end{bmatrix}^{-1} \begin{bmatrix} G_1 & G_3 \\ G_2 & G_4 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{2.9}$$

Thus, the transfer function matrix of the P+V canonical structure is:

$$G = \begin{bmatrix} 1 & -G_6 \\ -G_5 & 1 \end{bmatrix}^{-1} \begin{bmatrix} G_1 & G_3 \\ G_2 & G_4 \end{bmatrix} \tag{2.10}$$

Using eqn 2.10, a P+V structure of a process can be converted into a P-canonical form

and conversely a P-canonical structure can also be converted into a P+V form.

From the above three subsections, it can be concluded that a multivariable system can be represented by a P-canonical structure without any loss in generality.

In eqn 2.11, a *Multi-input Multi-output (MIMO)* system with *m-inputs, n-outputs* and distinct time-delays is represented in the P-canonical form, (also referred to as the Transfer function matrix (TFM) form).

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} G_{11}(z^{-1}) & G_{12}(z^{-1}) & \cdots & \cdots & G_{1m}(z^{-1}) \\ G_{21}(z^{-1}) & \cdots & \cdots & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ G_{n1}(z^{-1}) & \cdots & \cdots & \cdots & G_{nm}(z^{-1}) \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ \vdots \\ u_m \end{bmatrix} \tag{2.11}$$

where,

$$G_{ij}(z^{-1}) = \frac{B_{ij}(z^{-1})}{A_{ij}(z^{-1})} z^{-d_{ij}} \tag{2.12}$$

i = 1, 2, ...n (= number of outputs)

j = 1, 2, ...m (= number of inputs)

$$\begin{aligned} B_{ij}(z^{-1}) &= b_{ij,1}z^{-1} + b_{ij,2}z^{-2} + \cdots\cdots + b_{ij,\gamma_{ij}}z^{-\gamma_{ij}} \\ A_{ij}(z^{-1}) &= 1.0 + a_{ij,1}z^{-1} + \cdots\cdots + a_{ij,\nu_{ij}}z^{-\nu_{ij}} \end{aligned} \tag{2.13}$$

where, d_{ij} represents the delay in the (i,j)th element of the transfer function matrix

γ_{ij} = degree of the B-polynomial

v_{ij} = degree of the A-polynomial

Note the presence of distinct time-delays in each element of the transfer function matrix.

This is a very general form of representation for multivariable systems. However, it is not convenient to use for identification purposes. This is because, to identify the system parameters, the input-output data of the system need to be used sequentially as they become available, and to facilitate this, the system should be represented in an auto-regressive form or in an input-output difference equation form.

Example: A single input, single output (SISO) system is shown below in a transfer function form :

$$y(t) = G(z^{-1})u(t)$$

$$or \qquad (2.14)$$

$$y(t) = \frac{B(z^{-1})}{A(z^{-1})}u(t)$$

The auto-regressive model of this system will be as follows:

$$A(z^{-1})y(t) = B(z^{-1})u(t) \qquad (2.15)$$

This equivalent representation in auto-regressive form is trivial for SISO systems. But in MIMO systems, this equivalent representation is non-trivial because a matrix of transfer-function elements is involved. The following subsections discuss a simple transformation that allows one to decompose an n x m MIMO system into n MISO subsystems, each with m inputs.

2.1.4 Observable canonical state-space representation of MIMO systems with distinct time-delays

The MIMO system with distinct time-delays as represented in the transfer function matrix form in the previous section, can be transformed into an observable canonical state-space form. Kailath (1980) discusses the representation of multivariable systems by the observable canonical state-space form in the continuous domain. The same concept can be used for the representation of a process in the discrete domain. In the system described by eqn 2.11, because of the presence of a distinct time-delay in each element of the transfer function matrix, the above concept cannot be used directly. A modification of the structure is necessary to accommodate the distinct delays.

The MIMO system in eqn 2.11 is decomposed into a number of multi-input single-output (MISO) subsystems, with the number of MISO subsystems being equal to the number of outputs in the system. The time-delay in each element of the transfer function matrix is then included in the input vector. The MIMO system after decomposition into *n MISO subsystems* can be represented as follows:

$$y_1(t) = \begin{bmatrix} \dfrac{B_{11}(z^{-1})}{A_{11}(z^{-1})} & \dfrac{B_{12}(z^{-1})}{A_{12}(z^{-1})} & \cdots & \cdots & \cdots & \dfrac{B_{1m}(z^{-1})}{A_{1m}(z^{-1})} \end{bmatrix} \begin{bmatrix} u_1(t-d_{11}) \\ u_2(t-d_{12}) \\ \vdots \\ \vdots \\ \vdots \\ u_m(t-d_{1m}) \end{bmatrix} \qquad (2.16)$$

$$y_2(t) = \begin{bmatrix} \frac{B_{21}(z^{-1})}{A_{21}(z^{-1})} & \frac{B_{22}(z^{-1})}{A_{22}(z^{-1})} & \cdots & \cdots & \cdots & \frac{B_{2m}(z^{-1})}{A_{2m}(z^{-1})} \end{bmatrix} \begin{bmatrix} u_1(t-d_{21}) \\ u_2(t-d_{22}) \\ \vdots \\ \vdots \\ \vdots \\ u_m(t-d_{2m}) \end{bmatrix} \quad (2.17)$$

$$\begin{array}{cccc} : & : & : & : \\ : & : & : & : \\ : & : & : & : \end{array}$$

$$y_n(t) = \begin{bmatrix} \frac{B_{n1}(z^{-1})}{A_{n1}(z^{-1})} & \frac{B_{n2}(z^{-1})}{A_{n2}(z^{-1})} & \cdots & \cdots & \cdots & \frac{B_{nm}(z^{-1})}{A_{nm}(z^{-1})} \end{bmatrix} \begin{bmatrix} u_1(t-d_{n1}) \\ u_2(t-d_{n2}) \\ \vdots \\ \vdots \\ \vdots \\ u_m(t-d_{nm}) \end{bmatrix} \quad (2.18)$$

Each of the transfer function elements can now be represented in the observable canonical state-space form. As an example the representation of the element $G_{ij}(z^{-1})$ in eqn 2.11 is shown below.

$$G_{ij}(z^{-1}) = \frac{B_{ij}(z^{-1})}{A_{ij}(z^{-1})} z^{-d_{ij}} = \frac{b_{ij,1}z^{-1} + b_{ij,2}z^{-2} + \cdots\cdots + b_{ij,\gamma_{ij}}z^{-\gamma_{ij}}}{1.0 + a_{ij,1}z^{-1} + a_{ij,2}z^{-2} + \cdots\cdots + a_{ij,\nu_{ij}}z^{-\nu_{ij}}} z^{-d_{ij}} \quad (2.19)$$

where,

γ_{ij} = degree of the B-polynomial in the (i,j)th element of the TFM

ν_{ij} = degree of the A-polynomial in the (i,j)th element of the TFM

With this decomposition, the state-space representation of the i,jth element, G_{ij}, in the observable canonical form is given by:

$$\underline{X}_{ij}(t+1) = \begin{bmatrix} -a_{ij,1} & 1 & 0 & \cdots & \cdots & 0 \\ -a_{ij,2} & 0 & 1 & \ddots & \cdots & \vdots \\ \vdots & \vdots & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & 0 \\ \vdots & \vdots & \vdots & \cdots & \ddots & 1 \\ -a_{ij,\nu_{ij}} & 0 & 0 & \cdots & \cdots & 0 \end{bmatrix} \underline{X}_{ij}(t) + \begin{bmatrix} b_{ij,1} \\ b_{ij,2} \\ \vdots \\ \vdots \\ \vdots \\ b_{ij,\gamma_{ij}} \end{bmatrix} u_j(t-d_{ij}) \qquad (2.20)$$

or

$$\underline{X}_{ij}(t+1) = \underline{\underline{A}}_{ij}\ \underline{X}_{ij}(t) + \underline{B}_{ij}\ u_j(t-d_{ij}) \qquad (2.21)$$

where,

$\underline{X}_{ij}$ = state vector with ν_{ij} states of the (i,j)th element of the TFM

$\underline{\underline{A}}_{ij}$ = state system matrix of the (i,j)th element of the TFM

$\underline{B}_{ij}$ = state input vector of the (i,j)th element of the TFM

If, $\gamma_{ij} < \nu_{ij}$, then $\underline{B}_{ij}$ will have $(\gamma_{ij} - \nu_{ij})$ zeros after the last non-zero element in $\underline{B}_{ij}$. Since G_{ij} is the j^{th} element of the i^{th} MISO subsystem, it is not possible to write the output equation of the system at this stage. It can be written only when the state equations of all the elements of the i^{th} MISO subsystem in eqn 2.13 are complete.

Similarly, all the elements in eqn 2.13 can be expressed in the state-space form. By augmenting these structures, the state-space form of ith MISO subsystem can be obtained as follows:

$$\begin{bmatrix} \underline{X}_{i1}(t+1) \\ \underline{X}_{i2}(t+1) \\ \underline{X}_{i3}(t+1) \\ \vdots \\ \vdots \\ \underline{X}_{im}(t+1) \end{bmatrix} = \begin{bmatrix} \underline{\underline{A}}_{i1} & \underline{\underline{0}} & \cdots & \cdots & \cdots & \cdots & \underline{\underline{0}} \\ \underline{\underline{0}} & \underline{\underline{A}}_{i2} & \underline{\underline{0}} & \cdots & \cdots & \cdots & \vdots \\ \underline{\underline{0}} & \underline{\underline{0}} & \underline{\underline{A}}_{i3} & \underline{\underline{0}} & \cdots & \cdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \cdots & \ddots & \ddots & \underline{\underline{0}} \\ \underline{\underline{0}} & \cdots & \cdots & \cdots & \cdots & \underline{\underline{0}} & \underline{\underline{A}}_{im} \end{bmatrix} \begin{bmatrix} \underline{X}_{i1}(t) \\ \underline{X}_{i2}(t) \\ \underline{X}_{i3}(t) \\ \vdots \\ \vdots \\ \underline{X}_{im}(t) \end{bmatrix} +$$

$$\begin{bmatrix} \underline{B}_{i1} & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & \underline{B}_{i2} & 0 & \cdots & \cdots & \cdots & \vdots \\ 0 & 0 & \underline{B}_{i3} & 0 & \cdots & \cdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 & \underline{B}_{im} \end{bmatrix} \begin{bmatrix} u_1(t-d_{i1}) \\ u_2(t-d_{i2}) \\ u_3(t-d_{i3}) \\ \vdots \\ \vdots \\ \vdots \\ u_m(t-d_{im}) \end{bmatrix} \qquad (2.22)$$

$$y_i(t) = [\underset{\uparrow}{1}\ 0 \cdots 0,\ \underset{\uparrow}{1}\ 0 \cdots 0,\ \cdots\ \cdots \cdots,\ \underset{\uparrow}{1}\ 0 \cdots 0] \begin{bmatrix} \underline{X}_{i1}(t) \\ \underline{X}_{i2}(t) \\ \vdots \\ \vdots \\ \vdots \\ \underline{X}_{im}(t) \end{bmatrix} \qquad (2.23)$$

$1 \qquad \nu_{i1}+1 \qquad \nu_{i1}+\nu_{i(m-1)}+1$

or compactly:

$$\begin{aligned} \underline{X}_i(t+1) &= \underline{\underline{A}}_i \underline{X}_i(t) + \underline{B}_i U_i \\ y_i(t) &= \underline{C}_i \underline{X}_i(t) \end{aligned} \tag{2.24}$$

where, the number of states in the i^{th} MISO subsystem is given as follows:

$$s_i = \sum_{j=1}^{m} v_{ij} \tag{2.25}$$

Similarly, each MISO subsystem can be represented in observable canonical state-space form. By augmenting the structures of the n MISO subsystems, the MIMO system in eqn 2.11 can be represented in its observable canonical state-space form as follows:

$$\begin{bmatrix} \underline{X}_1(t+1) \\ \underline{X}_2(t+1) \\ \vdots \\ \vdots \\ \vdots \\ \underline{X}_n(t+1) \end{bmatrix} = \begin{bmatrix} \underline{\underline{A}}_1 & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & \underline{\underline{A}}_2 & 0 & \cdots & \cdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & \cdots & \cdots & 0 & \underline{\underline{A}}_n \end{bmatrix} \begin{bmatrix} \underline{X}_1(t) \\ \underline{X}_2(t) \\ \vdots \\ \vdots \\ \vdots \\ \underline{X}_n(t) \end{bmatrix} +$$

$$\begin{bmatrix} \underline{\underline{B}}_1 & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & \underline{\underline{B}}_2 & 0 & \cdots & \cdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & \cdots & \cdots & 0 & \underline{\underline{B}}_n \end{bmatrix} \begin{bmatrix} \underline{U}_1(t) \\ \underline{U}_2(t) \\ \vdots \\ \vdots \\ \vdots \\ \underline{U}_n(t) \end{bmatrix} \tag{2.26}$$

$$\begin{bmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ \vdots \\ \vdots \\ y_n(t) \end{bmatrix} = \begin{bmatrix} \underline{\underline{C_1}} & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & \underline{\underline{C_2}} & 0 & \cdots & \cdots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \cdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \cdots & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & \cdots & 0 & \underline{\underline{C_n}} \end{bmatrix} \begin{bmatrix} \underline{X_1}(t) \\ \underline{X_2}(t) \\ \vdots \\ \vdots \\ \vdots \\ \underline{X_n}(t) \end{bmatrix} \tag{2.27}$$

or more compactly as

$$\begin{aligned} \underline{X}(t+1) &= \underline{\underline{F}}\,\underline{X}(t) + \underline{\underline{G}}\,\underline{U} \\ \underline{Y}(t) &= \underline{\underline{H}}\,\underline{X}(t) \end{aligned} \tag{2.28}$$

where, the total number of states in the MIMO system is given as follows:

$$S = \sum_{i=1}^{n} s_i = \sum_{i=1}^{n} \sum_{j=1}^{m} v_{ij} \tag{2.29}$$

$\underline{\mathbf{F}}$ = state system matrix of dimension S x S

$\underline{\underline{\mathbf{G}}}$ = state input matrix of dimension S x m.n

$\underline{\underline{\mathbf{H}}}$ = state output matrix of dimension n x S

$\underline{\mathbf{X}}$ = state vector of length S

$\underline{\mathbf{Y}}$ = Output vector of length n

$\underline{\mathbf{U}}$ = Input vector of length m.n

2.1.5 Deterministic Auto-Regressive Moving Average (DARMA) representation of MIMO systems with distinct time-delays

MIMO systems with distinct time-delays in the transfer function matrix form as shown in eqn 2.11 can be transformed into the following equivalent auto-regressive form:

$$\begin{bmatrix} p_1(z^{-1}) & 0 & \cdots & \cdots & 0 \\ 0 & p_2(z^{-1}) & 0 & \cdots & 0 \\ \vdots & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & p_n(z^{-1}) \end{bmatrix} \begin{bmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ \vdots \\ y_n(t) \end{bmatrix} =$$

$$\begin{bmatrix} q_{11}(z^{-1})z^{-d_{11}} & q_{12}(z^{-1})z^{-d_{12}} & \cdots & \cdots & \cdots & q_{1m}(z^{-1})z^{-d_{1m}} \\ q_{21}(z^{-1})z^{-d21} & \cdots & & \cdots & \cdots & q_{2m}(z^{-d_{2m}} \\ \vdots & \vdots & & \vdots & \vdots & \vdots \\ \vdots & \vdots & & \vdots & \vdots & \vdots \\ q_{n1}(z^{-1})z^{-d_{n1}} & \cdots & & \cdots & \cdots & q_{nm}(z^{-1})z^{-d_{nm}} \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ \vdots \\ u_m(t) \end{bmatrix} \qquad (2.33)$$

This special form of representation, in which P is a diagonal polynomial matrix, is very convenient for identification, because the same concept used in subsection 2.1.3 to decompose a MIMO system into n MISO subsystem can be used to facilitate the identification of model parameters.

In the following section, the representation of MIMO systems with distinct delays (as represented by eqn 2.33) is shown to be equivalent to the general P-canonical form of representation shown in eqn 2.11.

2.2 Parametric equivalence between the TFM form and the DARMA form of a MIMO system

In the previous section, different representations of a MIMO system with distinct delays have been discussed. It has been shown in sec 2.1.1 that a MIMO system represented in the TFM form is a general form of representation. Also the need to use the DARMA form of structural representation for a MIMO system has also been shown in section 2.1.4. In this section, the MIMO system in the TFM form (please see eqn 2.11) is transformed into the DARMA form (eqn. 2.33). It is also shown that the DARMA form that is obtained always has P-matrix in diagonal form.

This transformation is based on the work of Guidorzi (1975, 1981), in which the structural and parametric equivalence between discrete multivariable systems represented by state-space equations and input-output difference descriptions is discussed. To obtain a workable transformation using these results, the MIMO system described by eqn 2.11 is represented in an observable canonical state-space form. This has already been discussed in sec 2.1.3, where the system in eqn 2.11 was represented in an observable canonical state-space form (please see eqns 2.27 and 2.28).

The remaining transformation requires the following steps:

Let,

$$\underline{\underline{H^T}} = [\, h_1 \;\; h_2 \;\; \cdots \;\; \cdots \;\; h_n \,] \tag{2.34}$$

Construct the following vector sequences from F and H,

h_1, $\underline{\underline{F}}^T h_1$, $(\underline{\underline{F}}^T)^2 h_1$,....., h_n, $\underline{\underline{F}}^T h_n$, $(\underline{\underline{F}}^T)^2 h_n$,.....

Rearranging these sequences in the following order gives:

$h_1, h_2, \ldots\ldots, h_n, \underline{F}^T h_1, \underline{F}^T h_2, \ldots\ldots, \underline{F}^T h_n, (\underline{F}^T)^2 h_1, \ldots\ldots\ldots\ldots\ldots$

Let,

ρ_i denote the number of vectors from the sequences of h_i, then since the MIMO system is assumed to be completely observable, it follows that:

$\rho_1 + \rho_2 + \ldots\ldots\ldots\ldots\ldots + \rho_n = S$ (total number of states of the system)

The set of integers $\rho_1, \rho_2, \ldots\ldots\ldots, \rho_n$ completely defines the structures of F and H and is invariant with respect to a change in co-ordinates. A change of co-ordinates leads to a new state vector,

$$\underline{\underline{W}} = \underline{\underline{T}}\,\underline{X} \tag{2.35}$$

where,

$$\underline{\underline{T}}^T = \left[h_1 \quad \underline{\underline{F}}^T h_1 \quad \ldots \quad (\underline{\underline{F}}^T)^{(\rho_1 - 1)} h_1 \quad \ldots\ldots\ldots \quad h_n \quad \underline{\underline{F}}^T h_n \quad \ldots \quad (\underline{\underline{F}}^T)^{(\rho_n - 1)} h_n\right] \tag{2.36}$$

$\underline{\underline{T}}$ is the transformation matrix of dimension S x S. Since $\underline{\underline{F}}$ and $\underline{\underline{H}}$ are in the block diagonal form, $\underline{\underline{T}}$ will also be in the block diagonal form.

The new system of equations after the transformation will be as follows:

$$\begin{aligned} \underline{\underline{W}}(t+1) &= \underline{\underline{A}}\,\underline{\underline{W}}(t) + \underline{\underline{B}}\,\underline{U}(t) \\ \underline{Y}(t) &= \underline{\underline{C}}\,\underline{\underline{W}}(t) \end{aligned} \tag{2.37}$$

where matrices $\underline{\underline{A}}$, $\underline{\underline{B}}$ and $\underline{\underline{C}}$ are of the following form:

$$\underline{\underline{A}} = \underline{\underline{T}}\,\underline{\underline{F}}\,\underline{\underline{T}}^{-1} = \{A_{ij}\} \tag{2.38}$$

i,j = 1,2,.....n

where, the block-diagonal components, A_{ii} of $\underline{\underline{A}}$ are :

$$A_{ii} = \begin{bmatrix} 0 & 1 & 0 & \cdots & \cdots & 0 \\ 0 & 0 & 1 & \ddots & \cdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & \cdots & 0 & 1 \\ a_{ii,1} & a_{ii,2} & \cdots & \cdots & \cdots & a_{ii,\nu_i} \end{bmatrix} \quad A_{ij} = \begin{bmatrix} 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ a_{ij,1} & \cdots & a_{ij,\nu_j} & 0 & \cdots & 0 \end{bmatrix} \tag{2.39}$$

the components of $\underline{\underline{B}}$ are :

$$\underline{\underline{B}} = \underline{\underline{T}}\underline{\underline{G}} = \begin{bmatrix} b_1^T \\ b_2^T \\ \vdots \\ \vdots \\ b_s^T \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} & \cdots & \cdots & b_{1,m.n} \\ b_{21} & \cdots & \cdots & \cdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ b_{s1} & b_{s2} & \cdots & \cdots & b_{s,m.n} \end{bmatrix} \tag{2.40}$$

and the components of $\underline{\underline{C}}$ are :

$$\underline{\underline{C}} = \underline{\underline{H}}\underline{\underline{T}}^{-1} = \begin{bmatrix} 1 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & \cdots & \cdots & 0 & 1 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \\ 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 & 1 & 0 & \cdots & 0 \end{bmatrix} \tag{2.41}$$

$$\begin{matrix} \uparrow & & \uparrow & & \uparrow \\ 1 & & (\nu_1+1) & & (\nu_1+\ldots.+\nu_{n-1}+1) \end{matrix}$$

Using the results from Guidorzi (1975), the system can be transformed into an input-output difference equation form as follows:

$$\begin{bmatrix} p_{11}(z) & \cdots & \cdots & \cdots & p_{1n}(z) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ p_{n1}(z) & \cdots & \cdots & \cdots & p_{nn}(z) \end{bmatrix} \begin{bmatrix} y_1(t) \\ \vdots \\ \vdots \\ \vdots \\ y_n(t) \end{bmatrix} = \begin{bmatrix} q_{11}(z) & q_{12}(z) & \cdots & \cdots & \cdots & q_{1,m.n}(z) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ q_{n1}(z) & \cdots & \cdots & \cdots & \cdots & q_{n,m.n}(z) \end{bmatrix} \begin{bmatrix} u_1(t-d_{11}) \\ \vdots \\ u_m(t-d_{1m}) \\ \vdots \\ \vdots \\ \vdots \\ u_1(t-d_{n1}) \\ \vdots \\ u_m(t-d_{mn}) \end{bmatrix} \quad (2.42)$$

where the polynomials matrix elements are defined as follows :

$$\begin{aligned} p_{ii}(z) &= z^{v_i} - a_{ii,v_i} z^{v_i-1} - \cdots\cdots - a_{ii,2} z - a_{ii,1} \\ p_{ij}(z) &= -a_{ij,v_j} z^{v_j-1} - \cdots\cdots - a_{ij,2} - a_{ij,1} \\ q_{ij}(z) &= \beta_{(v_1+\cdots+v_i)j} z^{v_i-1} + \cdots\cdots + \beta_{(v_1+\cdots+v_{i-1}+2)j} z + \beta_{(v_1+\cdots+v_{i-1}+1)j} \end{aligned} \quad (2.43)$$

The coefficients, β, of the q_{ij} polynomials are the elements of the matrix $\underline{\underline{\beta}}$ which is defined as:

$$\underline{\underline{\beta}} = \underline{\underline{M}}\,\underline{\underline{B}} = \begin{bmatrix} \beta_{11} & \cdots & \cdots & \beta_{1,m.n} \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ \beta_{s1} & \cdots & \cdots & \beta_{s,m.n} \end{bmatrix} \quad (2.44)$$

and $\underline{\underline{M}}$ is :

$$
M = \begin{bmatrix}
-a_{11,2} & -a_{11,3} & \cdots & -a_{11,v_1} & 1 & \cdots\cdots & -a_{1n,2} & \cdots & \cdots & -a_{1n,v_n} & 0 \\
-a_{11,3} & -a_{11,4} & \cdots & 1 & 0 & \cdots\cdots & -a_{1n,3} & \cdots & \cdots & & \vdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \cdots\cdots & \vdots & \vdots & & & \vdots \\
\vdots & \vdots & & & & \cdots\cdots & -a_{1m,v_n} & & & & \vdots \\
-a_{11,v_1} & 1 & & & & \cdots\cdots & 0 & & & & \vdots \\
1 & & & & & \cdots\cdots & 0 & & & & 0 \\
 & & & & & & & & & & \\
-a_{n1,2} & \cdots & \cdots & -a_{n1,v_{n1}} & 0 & \cdots\cdots & -a_{nn,2} & \cdots & \cdots & -a_{nn,v_n} & 1 \\
-a_{n1,3} & \cdots & \cdots & 0 & & \cdots\cdots & -a_{nn,3} & \cdots & & & \\
\vdots & \vdots & & & & \cdots\cdots & \vdots & & & & \\
-a_{n1,v_{n1}} & 0 & & & & \cdots\cdots & -a_{nn,v_n} & & & & \\
0 & & & & & \cdots\cdots & 1 & & & &
\end{bmatrix} \qquad (2.41)
$$

Since **T** and **F** are in block diagonal form, **A** will also be in a block diagonal form. Thus, **M** will also be in a block diagonal form. Since **G** is in a block diagonal form, **B** will also be in a block diagonal form, and $\underline{\beta}$ inturn will also be in a block diagonal form. Since all the off-diagonal polynomials in the **P** matrix are in terms of the off-diagonal blocks of the block-diagonal **A** matrix, this implies that **P** will be a diagonal polynomial matrix with all of the off-diagonal polynomials in **P** matrix equal to zero.

Also, since **B** and **M** are in block diagonal form, $\underline{\beta}$ will be in the block diagonal form. Thus the **Q** matrix whose elements are obtained from the $\underline{\beta}$ matrix will have a special form. This is shown in terms of the forward shift operator (z) as follows:

$$\underline{\underline{Q}} = \begin{bmatrix} q_{11} & q_{12} & \cdots & q_{1m} & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & 0 & \cdots & 0 & q_{21} & q_{22} & \cdots & q_{2m} & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & 0 & \cdots & \cdots & \cdots & \ddots & \ddots & \ddots & \ddots & \cdots & \cdots & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & q_{n1} & q_{n2} & \cdots & q_{nm} \end{bmatrix} \tag{2.42}$$

Thus the original MIMO system in a DARMA (Deterministic Auto-Regressive Moving Average) form can be expressed as follows: (from eqn 2.38, 2.39, 2.40, 2.41, 2.42)

$$\begin{bmatrix} p_{11} & 0 & \cdots & \cdots & 0 \\ 0 & p_{22} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & p_{nn} \end{bmatrix} \begin{bmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ \vdots \\ y_n(t) \end{bmatrix} =$$

$$\begin{bmatrix} q_{11} & q_{12} & \cdots & q_{1m} & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & 0 & \cdots & 0 & q_{21} & q_{22} & \cdots & q_{2m} & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & 0 & \cdots & \cdots & \cdots & \ddots & \ddots & \ddots & \ddots & \cdots & \cdots & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & q_{n1} & q_{n2} & \cdots & q_{nm} \end{bmatrix} \begin{bmatrix} u_1(t-d_{11}) \\ u_2(t-d_{12}) \\ \vdots \\ u_m(t-d_{1m}) \\ u_1(t-d_{21}) \\ \vdots \\ u_m(t-d_{2m}) \\ \vdots \\ \vdots \\ u_1(t-d_{n1}) \\ u_2(t-d_{n2}) \\ \vdots \\ u_m(t-d_{nm}) \end{bmatrix} \tag{2.47}$$

or in a compact form as:

$$\begin{bmatrix} p_1 & 0 & \cdots & \cdots & 0 \\ 0 & p_2 & 0 & \cdots & 0 \\ \vdots & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & p_n \end{bmatrix} \begin{bmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ \vdots \\ y_n(t) \end{bmatrix} = \begin{bmatrix} q_{11}z^{-d_{11}} & q_{12}z^{-d_{12}} & \cdots & \cdots & \cdots & q_{1m}z^{-d_{1m}} \\ q_{21}z^{-d21} & \cdots & \cdots & \cdots & \cdots & q_{2m}z^{-d_{2m}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ q_{n1}z^{-d_{n1}} & \cdots & \cdots & \cdots & \cdots & q_{nm}z^{-d_{nm}} \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ \vdots \\ \vdots \\ u_m(t) \end{bmatrix} \qquad (2.48)$$

Here the p and the q polynomials are in terms of the forward shift operator (z), but the delays are in terms of the backward shift operator. To transform the equation into the backward shift operator (z^{-1}), multiply both sides by,

$$V = \begin{bmatrix} z^{-v_1} & 0 & \cdots & \cdots & 0 \\ 0 & z^{-v_2} & 0 & \cdots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \cdots & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & z^{-v_n} \end{bmatrix} \qquad (2.49)$$

where, $v_1, v_2, \ldots\ldots, v_n$ are the highest degree of the polynomial in the respective rows of the composite matrix [P,Q].

If the highest degree is equal to the degree of the polynomial in the P matrix, then the polynomials in the P-matrix will be monic, i.e. their first element will be 1.

Thus eqn 2.48 becomes :

$$\begin{bmatrix} p_1(z^{-1}) & 0 & \cdots & \cdots & 0 \\ 0 & p_2(z^{-1}) & 0 & \cdots & 0 \\ \vdots & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & p_n(z^{-1}) \end{bmatrix} \begin{bmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ \vdots \\ y_n(t) \end{bmatrix} =$$

$$\begin{bmatrix} q_{11}(z^{-1})z^{-d_{11}} & q_{12}(z^{-1})z^{-d_{12}} & \cdots & \cdots & \cdots & q_{1m}(z^{-1})z^{-d_{1m}} \\ q_{21}(z^{-1})z^{-d21} & \cdots & & \cdots & \cdots & q_{2m}(z^{-d_{2m}} \\ \vdots & \vdots & & \vdots & \vdots & \vdots \\ \vdots & \vdots & & \vdots & \vdots & \vdots \\ q_{n1}(z^{-1})z^{-d_{n1}} & \cdots & & \cdots & \cdots & q_{nm}(z^{-1})z^{-d_{nm}} \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ \vdots \\ \vdots \\ u_m(t) \end{bmatrix} \quad (2.50)$$

The above analysis has shown that an n x m MIMO system with distinct time-delays in the transfer function matrix form can be transformed into a DARMA form in which the $\underline{\mathbf{P}}(z^{-1})$ matrix will always be a diagonal polynomial square matrix (of dimension n) and the $\underline{\mathbf{Q}}(z^{-1})$ will be a full, polynomial matrix (of dimension n x m).

2.2.1 An Illustrative Example

Consider the 2 input 2 output system with distinct time-delays in each of its transfer function elements:

$$\begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} = \begin{bmatrix} \frac{0.2z^{-1}}{1-0.8z^{-1}}z^{-2} & \frac{0.3z^{-1}}{1-0.7z^{-1}}z^{-3} \\ \frac{0.5z^{-1}}{1-0.6z^{-1}}z^{-4} & \frac{0.6z^{-1}}{1-0.9z^{-1}}z^{-5} \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix} \quad (2.51)$$

This system can be represented in the observable canonical state-space form as follows:

$$\begin{bmatrix} x_1(t+1) \\ x_2(t+1) \\ x_3(t+1) \\ x_4(t+1) \end{bmatrix} = \begin{bmatrix} 0.8 & 0 & 0 & 0 \\ 0 & 0.7 & 0 & 0 \\ 0 & 0 & 0.6 & 0 \\ 0 & 0 & 0 & 0.9 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix} + \begin{bmatrix} 0.2 & 0 & 0 & 0 \\ 0 & 0.3 & 0 & 0 \\ 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0.6 \end{bmatrix} \begin{bmatrix} u_1(t-2) \\ u_2(t-3) \\ u_1(t-4) \\ u_2(t-5) \end{bmatrix} \tag{2.52}$$

$$\begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix} \tag{2.53}$$

or in a more compact form as:

$$\begin{aligned} X(t+1) &= F\,X(t) + G\,U(t) \\ Y(t) &= H\,X(t) \end{aligned} \tag{2.54}$$

where,

$$H = \begin{bmatrix} h_1^T \\ h_2^T \end{bmatrix} \tag{2.55}$$

The transformation matrix T in terms of F and H is as follows:

$$T = \left[h_1^T,\ F^T h_1^T,\ h_2^T,\ F^T h_2^T\right] = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0.8 & 0.7 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0.6 & 0.9 \end{bmatrix} \tag{2.56}$$

The new transformed state space equation is :

$$W(t+1) = A\,W(t) + B\,U(t)$$
$$Y(t) = C\,W(t) \tag{2.57}$$

where,

$W(t) = T\,X(t)$

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -0.56 & 1.5 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -0.54 & 1.5 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ a_{11,1} & a_{11,2} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & a_{22,1} & a_{22,2} \end{bmatrix} \tag{2.58}$$

$$B = T\,G = \begin{bmatrix} 0.2 & 0.3 & 0 & 0 \\ 0.16 & 0.21 & 0 & 0 \\ 0 & 0 & 0.5 & 0.6 \\ 0 & 0 & 0.3 & 0.54 \end{bmatrix} \tag{2.59}$$

$$C = H\,T^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{2.60}$$

The system in the DARMA form will be

$$\begin{bmatrix} p_{11}(z) & p_{12}(z) \\ p_{21}(z) & p_{22}(z) \end{bmatrix} \begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} = \begin{bmatrix} q_{11}(z) & q_{12}(z) & q_{13}(z) & q_{14}(z) \\ q_{21}(z) & q_{22}(z) & q_{23}(z) & q_{24}(z) \end{bmatrix} \begin{bmatrix} u_1(t-2) \\ u_2(t-3) \\ u_1(t-4) \\ u_2(t-5) \end{bmatrix} \tag{2.61}$$

where,

$p_{11}(z) = z^2 - a_{11,2}z - a_{11,1} = z^2 - 1.5z + 0.56$ $\quad p_{12}(z) = -a_{12,2}z - a_{12,1} = 0$

$p_{21}(z) = -a_{21,2}z - a_{21,1} = 0$ $\quad p_{22}(z) = z^2 - a_{22,2}z - a_{22,1} = z^2 - 1.5z + 0.54$

and

$q_{11}(z) = \beta_{21}z+\beta_{11} = 0.2z-0.14$ $\quad$ $q_{12}(z) = \beta_{22}z+\beta_{12} = 0.3z-0.24$

$q_{13}(z) = \beta_{23}z+\beta_{13} = 0$ $\quad$ $q_{14}(z) = \beta_{24}z+\beta_{14} = 0$

$q_{21}(z) = \beta_{41}z+\beta_{31} = 0$ $\quad$ $q_{22}(z) = \beta_{42}z+\beta_{32} = 0$

$q_{23}(z) = \beta_{43}z+\beta_{33} = 0.5z-0.45$ $\quad$ $q_{24}(z) = \beta_{44}z+\beta_{34} = 0.6z-0.36$

where,

$$\underline{B} = M\,B, \qquad with$$

$$M = \begin{bmatrix} -a_{11,2} & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & -a_{22,2} & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} -1,5 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & -1.5 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{2.62}$$

$$and \quad \underline{B} = \begin{bmatrix} \beta_{11} & \beta_{12} & \beta_{13} & \beta_{14} \\ \beta_{21} & \beta_{22} & \beta_{23} & \beta_{24} \\ \beta_{31} & \beta_{32} & \beta_{33} & \beta_{34} \\ \beta_{41} & \beta_{42} & \beta_{43} & \beta_{44} \end{bmatrix} = \begin{bmatrix} -0.14 & -0.24 & 0 & 0 \\ 0.2 & 0.3 & 0 & 0 \\ 0 & 0 & -0.45 & -0.36 \\ 0 & 0 & 0.5 & 0.6 \end{bmatrix} \tag{2.63}$$

Thus, the eqn 2.61 can be written as:

$$\begin{bmatrix} z^2-1.5z+0.56 & 0 \\ 0 & z^2-1.5z+0.54 \end{bmatrix} \begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} =$$

$$\begin{bmatrix} 0.2z-0.14 & 0.3z-0.24 & 0 & 0 \\ 0 & 0 & 0.5z-0.45 & 0.6z-0.36 \end{bmatrix} \begin{bmatrix} u_1(t-2) \\ u_2(t-3) \\ u_1(t-4) \\ u_2(t-5) \end{bmatrix} \tag{2.64}$$

Multiplying both sides by,

$$V = \begin{bmatrix} z^{-2} & 0 \\ 0 & z^{-2} \end{bmatrix} \tag{2.65}$$

converts eqn 2.64 to :

$$\begin{bmatrix} 1-1.5z^{-1}+0.56z^{-2} & 0 \\ 0 & 1-1.5z^{-1}+0.54z^{-2} \end{bmatrix} \begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} =$$

$$\begin{bmatrix} 0.2z^{-1}-0.14z^{-2} & 0.3z^{-1}-0.24z^{-2} & 0 & 0 \\ 0 & 0 & 0.5z^{-1}-0.45z^{-2} & 0.6z^{-1}-0.36z^{-2} \end{bmatrix} \begin{bmatrix} u_1(t-2) \\ u_2(t-3) \\ u_1(t-4) \\ u_2(t-5) \end{bmatrix} \tag{2.66}$$

The last equation can be expressed in:

$$\begin{bmatrix} 1-1.5z^{-1}+0.56z^{-2} & 0 \\ 0 & 1-1.5z^{-1}+0.54z^{-2} \end{bmatrix} \begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} =$$

$$\begin{bmatrix} (0.2z^{-1}-0.14z^{-2})z^{-2} & (0.3z^{-1}-0.24z^{-2})z^{-3} \\ (0.5z^{-1}-0.45z^{-2})z^{-4} & (0.6z^{-1}-0.36z^{-2})z^{-5} \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix} \tag{2.67}$$

Compare eqn 2.51 and eqn 2.67 which are the equivalent forms of the same system in the TFM form and the DARMA form respectively. It can be realized that although they are equivalent there is a difference in parameterization i.e. the system in the TFM form has 8 parameters and the system in the equivalent DARMA form has 12 parameters. Also it can be noticed that there is no need to go through this complete transformation each time, since the system in the DARMA form can be obtained just by cross-multiplying the polynomials in the elements of the TFM form. This is illustrated below and can be

applied to a general n x m systems.

The system in the TFM form in eqn 2.51, by considering the common denominator in each row of the matrix can be written as follows:

$$\begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} = \begin{bmatrix} \dfrac{(0.2z^{-1})\ (1-0.7z^{-1})}{(1-0.7z^{-1})(1-0.8z^{-1})} z^{-2} & \dfrac{(0.3z^{-1})\ (1-0.8z^{-1})}{(1-0.7z^{-1})(1-0.8z^{-1})} z^{-3} \\ \dfrac{(0.5z^{-1})\ (1-0.9z^{-1}}{(1-0.6z^{-1})(1-0.9z^{-1})} z^{-4} & \dfrac{(0.6z^{-1})\ (1-0.6z^{-1})}{(1-0.6z^{-1})(1-0.9z^{-1})} z^{-5} \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix} \tag{2.68}$$

or

$$\begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} = \begin{bmatrix} \dfrac{1}{(1-0.8z^{-1})(1-0.7z^{-1})} & 0 \\ 0 & \dfrac{1}{(1-0.6z^{-1})(1-0.9z^{-1})} \end{bmatrix} \times \begin{bmatrix} (0.2z^{-1})(1-0.7z^{-1})z^{-2} & (0.3z^{-1})(1-0.8z^{-1})z^{-3} \\ (0.5z^{-1})(1-0.9z^{-1})z^{-4} & (0.6z^{-1})(1-0.6z^{-1})z^{-5} \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix} \tag{2.69}$$

or

$$\begin{bmatrix} (1-0.8z^{-1})(1-0.7z^{-1}) & 0 \\ 0 & (1-0.6z^{-1})(1-0.9z^{-1}) \end{bmatrix} \begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} = \begin{bmatrix} (0.2z^{-1})(1-0.7z^{-1})z^{-2} & (0.3z^{-1})(1-0.8z^{-1})z^{-3} \\ (0.5z^{-1})(1-0.9z^{-1})z^{-4} & (0.6z^{-1})(1-0.6z^{-1})z^{-5} \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix} \tag{2.70}$$

This is exactly same as eqn 2.67. Thus the final result of this transformation is very straight-forward. In the process of transforming the system from TFM to DARMA form, it has been proved that the P matrix in the DARMA form will always be in the diagonal form.

In this chapter the various structures or modes of representing a MIMO system with distinct time-delays have been discussed. It has been proved that a MIMO system can always be transformed from its transfer function matrix form to an equivalent DARMA form in which the output polynomial matrix $P(z^{-1})$ is always a diagonal matrix.

Chapter 3

Identification of Multivariable systems

On-line determination of the process parameters (parameter estimation or identification) is an important component of an adaptive controller. Estimation of process parameters in a multivariable system is a multi-dimensional optimization problem and presents a challenging and non-trivial task even for relatively small dimensional plants. This chapter discusses various identification techniques to estimate the process model parameters and extends one of them for identification of parameters in a multivariable systems.

The previous chapter discussed various structures to represent multivariable systems. A structure of the model of a MIMO system which is convenient to use in identification is reproduced here for convenience.

$$P(z^{-1})\ Y(t) \;=\; Q(z^{-1})\ U(t) \tag{3.1}$$

In particular, $P(z^{-1})$ is a diagonal and square polynomial matrix and $Q(z^{-1})$ is a full polynomial matrix.

This chapter is organized as follows:

In the first section of this chapter, some of the popular recursive identification techniques such as basic Recursive Least Squares (RLS), Bierman's UDU^T factorization algorithm for identification, Augmented UDU^T (AUD) factorization algorithm which is an extension of the UDU^T factorization algorithm, are discussed. In the second section, the AUD factorization algorithm is extended to identify MIMO systems with distinct time-delays in each of its transfer function element. The performance of this algorithm in the identification of an openloop process is demonstrated using data from a continuous stirred tank heater which is configured as a 2 x 2 MIMO system. The MIMO system can be represented by two different models:

(i) The Auto-Regressive, Moving Average with an exogenous input (ARMAX) model and (ii) the Auto-Regressive, Integrated Moving Average with an exogenous input (ARIMAX) model. Identification is performed on both the model forms and the suitability of each model of each model is discussed. To retain the alertness of this algorithm in the absence of excitation, this algorithm is implemented with a variable forgetting factor, including a lower bound on the forgetting factor.

3.1 Recursive Identification techniques

In most chemical systems where adaptive control is used, on-line identification of a model is necessary while the system is in operation. The need for on-line model identification arises because the control decisions that are to be taken at the next sampling instant should be based on the most current information about the system. This can only

be captured by updating the model at every time instant. Such controllers in which the changes occurring in the system during normal operation are observed, estimated and utilized are called *adaptive controllers*. The on-line computation of the model must be done in such a way that the processing of the measurements from one sample can be completed within one sampling interval, otherwise the information flow is not possible. The *techniques* that comply with the above requirements are called *recursive identification techniques*, since the measured input-output data are processed recursively (sequentially) as they become available. Other commonly used terms for such techniques are called on-line or real-time identification or adaptive parameter estimation (Ljung 1987).

3.1.1 Models for Identification

Consider a dynamic system with input signal *u(t)* and output signal *y(t)*. Let these signals be sampled in discrete time t = 1, 2, 3, ···. Let the sampled values of these signals be related through the linear difference equation as follows:

$$y(t) + a_1 y(t-1) + \cdots\cdots + a_{na} y(t-na) = b_1 u(t-1) + \cdots\cdots + b_{nb} u(t-nb) + v(t) \qquad (3.2)$$

where $v(t)$ is a sequence of independent random variables with mean of zero, also popularly called as *white noise* (Ljung and Söderström, 1983). Eqn 3.2 can be expressed in terms of the backward difference operator (z^{-1}) as follows:

$$\{1 + a_1 z^{-1} + \cdots\cdots + a_{na} z^{-na}\} y(t) = \{b_0 + b_1 z^{-1} + \cdots\cdots + b_{nb} z^{-nb}\} u(t-1) + v(t) \qquad (3.3)$$

or

$$A(z^{-1})\, y(t) = B(z^{-1})\, u(t-1) + v(t) \tag{3.4}$$

The model in eqn 3.2 or 3.4 describes the dynamic relationship between the input and output signals. The left hand side of eqn 3.4 is commonly known as an *auto-regressive process of order na*. The first term in the right hand side of the eqn 3.4 is known as the *exogenous input*. Thus the model in eqn 3.4 is called as an Auto-Regressive model with an exogenous input (ARX).

If the term $v(t)$ is modelled as a moving average of a white noise sequence e(t) defined as:

$$\begin{aligned} v(t) &= C(z^{-1})e(t) \\ &\textit{where,} \\ C(z^{-1}) &= 1 + c_1 z^{-1} + \cdots\cdots + c_r z^{-r} \end{aligned} \tag{3.5}$$

the resulting model becomes

$$A(z^{-1})y(t) = B(z^{-1})u(t-1) + C(z^{-1})e(t) \tag{3.6}$$

This is known as an Auto-Regressive model with a Moving Average and an exogenous input (ARMAX).

If the term $v(t)$ is modelled as an integrated moving average of a white noise sequence e(t) defined as:

$$A(z^{-1})y(t) = B(z^{-1})u(t-1) + \frac{C(z^{-1})e(t)}{\Delta} \tag{3.7}$$

the resulting model will be:

$$A(z^{-1})y(t) = B(z^{-1}) + \frac{C(z^{-1})e(t)}{\Delta}$$

This is known as an Auto-Regressive model with an Integrated Moving Average and an exogenous input (ARIMAX).

3.1.2 Preliminaries for recursive identification

The model in eqn 3.2 or 3.4 can be expressed in terms of a *parameter vector*:

$$\theta^T = \{ a_1 \cdots a_{na}\ b_1\ b_2 \cdots b_{nb} \} \tag{3.9}$$

and a vector containing the input-output data called the *regressor*:

$$\phi^T(t) = \{ -y(t-1) \cdots -y(t-na)\ u(t-1) \cdots u(t-nb) \} \tag{3.10}$$

Then eqn 3.2 can be written as follows:

$$y(t) = \theta^T\phi(t) + v(t) \tag{3.11}$$

The parameter vector θ can be estimated from the input-output data in the regressor vector ϕ(t) and y(t) by minimizing the quadratic cost function of the prediction error:

$$J_N(\theta) = \frac{1}{N}\sum_{t=1}^{N} \alpha_t[y(t) - \theta^T\phi(t)]^2 \tag{3.12}$$

where,

t = 1, 2, 3, ..., N is the sampling instant.

α_t = sequence of positive numbers by which different weights can be associated with different observations (assumed to be 1 in the following case), Ljung and Söderström, (1983).

3.1.3 The Recursive Least-Squares technique

The Recursive least squares (RLS) algorithm is one of the most popular parametric identification method used for on-line process model estimation and adaptive control (Shah *et al.*, 1991). It is one of the members of a family of prediction error identification methods that are based on the minimization of the prediction error functions (Ljung and Soderstrom, 1983). This algorithm has been discussed in detail by Ljung (1987), Ljung and Soderstrom (1983), Astrom and Wittenmark (1989) and Shah *et al.*(1991).

The basic RLS algorithm can be obtained by minimizing the quadratic cost function of the prediction error shown in eqn 3.12. The derivation is as follows:

$$J_N(\theta) = \frac{1}{N}\sum_{t=1}^{N} [\, y(t)^2 - 2\theta^T\phi(t)y(t) + (\theta^T\phi(t))^2 \,] \tag{3.13}$$

$$\frac{\partial J_N}{\partial \theta} = \frac{2}{N}\sum_{t=1}^{N} [\, -\phi(t)y(t) + \phi(t)\phi(t)^T\theta \,] = 0 \tag{3.14}$$

this gives:

$$\left[\sum_{t=1}^{N} \phi(t)\phi(t)^T\right]\theta = \sum_{t=1}^{N} \phi(t)y(t) \tag{3.15}$$

$$\therefore \hat{\theta}(N) = \left[\sum_{t=1}^{N} \phi(t)\phi(t)^T\right]^{-1}\sum_{t=1}^{N} \phi(t)y(t) \tag{3.16}$$

Then for any sampling instant t, eqn 3.16 can be written as:

$$\hat{\theta}(t) = \left[\sum_{k=1}^{t} \phi(k)\phi(k)^T \right]^{-1} \sum_{k=1}^{t} \phi(k)y(k) \tag{3.17}$$

Defining the covariance matrix of the regressor as:

$$P(t)^{-1} = \sum_{k=1}^{t} \phi(k)\phi(k)^T$$

$$or \quad P(t)^{-1} = \sum_{k=1}^{t-1} \phi(k)\phi(k)^T + \phi(t)\phi(t)^T \tag{3.18}$$

or recursively as,

$$P(t)^{-1} = P(t-1)^{-1} + \phi(t)\phi(t)^T$$

then eqn 3.17 becomes

$$\hat{\theta}(t) = P(t) \sum_{k=1}^{t} \phi(k)y(k) \quad \Rightarrow \quad P^{-1}\hat{\theta}(t-1) = \sum_{k=1}^{t-1} \phi(k)y(k) \tag{3.19}$$

$$or \quad \hat{\theta}(t) = P(t) \left[\sum_{k=1}^{t-1} \phi(k)y(k) + \phi(t)y(t) \right] \tag{3.20}$$

or substituting eqn 3.19 in eqn 3.20,

$$\hat{\theta}(t) = P(t) \left[P(t-1)^{-1}\hat{\theta}(t-1) + \phi(t)y(t) \right]$$

and using the recursive form of eqn 3.18,

$$\hat{\theta}(t) = P(t) \left[\{ P(t)^{-1} - \phi(t)\phi(t)^T \} \hat{\theta}(t-1) + \phi(t)y(t) \right] \tag{3.21}$$

$$\hat{\theta}(t) = \hat{\theta}(t-1) + P(t)\phi(t) \left[y(t) - \phi(t)^T\hat{\theta}(t-1) \right]$$

The covariance matrix update is given by eqn 3.18 and the parameter update by eqn 3.21. But the algorithm will be computationally expensive since it requires the inversion of the covariance matrix P(t) at every sampling instant. This problem has been solved by

using the matrix inversion lemma (Ljung, 1987).

The covariance matrix update will be:

$$P(t) = P(t-1) - \frac{P(t-1)\phi(t)\phi(t)^T P(t-1)}{1+\phi(t)^T P(t-1)\phi(t)} \tag{3.22}$$

The Gain vector update will be:

$$K(t) = P(t)\phi(t) = \frac{P(t-1)\phi(t)}{1+\phi(t)^T P(t-1)\phi(t)} \tag{3.23}$$

The basic RLS algorithm can be summarized by eqns 3.21, 3.22 and 3.23.

The main difficulty with this basic RLS algorithm is that the covariance matrix would gradually decay to a small value and therefore the algorithm does not retain its alertness. Shah *et al.*(1991) discusses several variants of RLS which retain the alertness of the algorithm. They are summarized as follows:

1. **RLS with exponential data weighting** in which a forgetting factor λ is used to give more weight to the most recent data. This parameter can be kept constant or variable. This variant of RLS is discussed in the following subsection.
2. **RLS with covariance resetting** in which the decay of the covariance matrix P(t) is reversed by increasing the elements of P(t) by a non-zero value at various times. Goodwin *et al.*(1983) provide a convergence analysis of this algorithm.
3. **RLS with matrix regularization** in which the idea of covariance resetting is combined with a guaranteed lower and upper bounds on the covariance matrix. This variant of RLS was proposed by Praly (1983) and was modified by Ortega *et al.*(1985) for its use in robust adaptive control.

4. **RLS with constant trace** which was proposed by Irving (1979) and recently modified by Goodwin *et al.*(1985). This algorithm ensures a constant trace of the covariance matrix.

3.1.3.1 RLS with exponential data weighting

The cost function of the prediction error in eqn 3.12 with exponential data weighting is incorporated with a forgetting factor λ as follows:

$$J_N(\theta) = \frac{1}{N}\sum_{t=1}^{N} \lambda^{N-t}\,\alpha_t[y(t)-\theta^T\phi(t)]^2 \tag{3.24}$$

The value of $\lambda < 1$ means that the cost function in eqn 3.24 is minimized with more heavier weighting on the more recent prediction errors.

From eqn 3.24, eqns 3.21, 3.22 and 3.23 are modified as follows:

$$\hat{\theta}(t) = \hat{\theta}(t-1) + P(t)\phi(t)\left[y(t) - \phi(t)^T\hat{\theta}(t-1)\right] \tag{3.25}$$

$$P(t) = \left[P(t-1) - \frac{P(t-1)\phi(t)\phi(t)^T P(t-1)}{\lambda + \phi(t)^T P(t-1)\phi(t)} \right]\frac{1}{\lambda} \tag{3.26}$$

$$K(t) = P(t)\phi(t) = \frac{P(t-1)\phi(t)}{\lambda + \phi(t)^T P(t-1)\phi(t)} \tag{3.27}$$

The length of past information used by the algorithm is determined by $1/(1-\lambda)$. Thus, a smaller value of λ means very fast forgetting of the older data, which can make the parameter estimates uncertain, but a large value of λ means slower forgetting of the older data will make the estimation of rapidly varying parameters difficult (Shah *et al.*, 1991). Since λ is in the denominator of the equation for P(t), the value of the covariance matrix

P(t) can grow exponentially in the absence of any excitation for fixed λ and a small change in the regressor ϕ(t) at a future time can lead to large changes in the parameter values. Fortescue *et al.*(1981) have shown that by using the idea of variable forgetting factor, the covariance matrix value can be kept from growing exponentially large. Thus, the parameters can also be kept bounded, although their variances can become large within the same interval. The time varying forgetting factor is adjusted so that $\lambda(t) \rightarrow 1$ when the prediction error is small and $\lambda(t) \rightarrow$ a small value if the prediction error is large. For the variable forgetting factor algorithm the eqns 3.26 and 3.27 are replaced by the following:

$$K(t) = P(t)\phi(t) = \frac{P(t-1)\phi(t)}{1+\phi(t)^T P(t-1)\phi(t)} \tag{3.28}$$

$$\lambda(t) = 1 - \frac{\left[y(t) - \hat{\theta}(t-1)^T\phi(t)\right]^2}{\left[\sigma\{1+\phi(t)^T P(t-1)\phi(t)\}\right]} \tag{3.29}$$

$$P(t) = \left[P(t-1) - \frac{P(t-1)\phi(t)\phi(t)^T P(t-1)}{\lambda(t)+\phi(t)^T P(t-1)\phi(t)}\right]\frac{1}{\lambda(t)} \tag{3.30}$$

if the trace of P(t) $\leq$ C (some user specified upper bound for the value of trace)

else, $\lambda(t) = 1$ and P(t) is not inflated by $1/\lambda(t)$.

3.1.4 UDU^T Factorization algorithm

The UDU^T factorization (UD) algorithm was proposed by Bierman (1977). It overcomes the numerical problems encountered in recursive least-squares (RLS) which was presented in the previous section. RLS has poor numerical performance when

implemented on computers with finite accuracy (Clarke, 1981). This can be explained from the eqn. 3.22 where the covariance matrix P(t) is updated from P(t-1) using a potentially unstable formula, and thus there is no guarantee that it will always be positive definite. The UD algorithm has been discussed in detail by Thornton and Bierman (1980) and Ljung and Soderstorm (1983).

The problem of numerical instability in RLS is avoided here by decomposing the covariance matrix P(t) as follows:

$$P(t) = U(t)D(t)U^T(t) \tag{3.31}$$

Where U(t) is the upper-triangular matrix and D(t) is the diagonal matrix. Then instead of updating P(t) at every time step (please see eqn 3.22), the matrix factors U(t) and D(t) are updated. This decomposition guarantees the positive-definiteness of the covariance matrix P(t), and thus accurate identification and robustness can be attained (Bierman, 1977; Ljung, 1985).

By applying eqn 3.31 to eqn 3.22, it is modified as follows:

$$U(t)D(t)U^T(t) = \left[U(t-1)D(t-1)U^T(t-1) - \frac{U(t-1)D(t-1)U^T(t-1)\phi(t)\phi^T(t)U(t-1)D(t-1)U^T(t-1)}{\lambda(t)+\phi^T(t)U(t-1)D(t-1)U^T(t-1)\phi(t)}\right]\frac{1}{\lambda(t)} \tag{3.32}$$

let,

$$f(t) = U^T(t-1)\phi(t) \tag{3.33}$$

$$g(t) = D(t-1)f(t) = D(t-1)U^T(t-1)\phi(t)$$

$$\beta(t) = \lambda(t) + \phi^T(t)U(t-1)D(t-1)U^T(t-1)\phi(t) = \lambda(t) + f^T(t)g(t)$$

Then, eqn 3.32 becomes,

$$U(t)D(t)U^T(t) = \left[U(t-1)D(t-1)U^T(t-1) - \frac{U(t-1)g(t)g^T(t)U^T(t-1)}{\beta(t)}\right]\frac{1}{\lambda(t)} \quad (3.34)$$

or

$$U(t)D(t)U^T(t) = U(t-1)\left[D(t-1) - \frac{g(t)g^T(t)}{\beta(t)}\right]\frac{U^T(t-1)}{\lambda(t)} \quad (3.35)$$

After introducing the basic steps, the complete algorithm in stepwise manner is presented below (for mathematical details please refer to Ljung and Soderstorm (1983)).

Procedure for Bierman's UD factorization algorithm (Bierman, 1977)

Initialize the algorithm with $P(0) = U(0)D(0)U^T(0) = \sigma^2 I$, where σ is a large integer. i.e. $U(0) = I$ and $D(0) = \sigma^2 I$. At time step $t > 0$, construct the data vector as in eqn 3.10, and update U(t-1) and D(t-1) by performing steps 1-7.

- **Step 1**

  Compute

$$f = U^T(t-1)\phi(t)$$
$$g = D(t-1)f$$
$$\beta_0 = \lambda(t)$$

- **Step 2**

  For j = 1,.....,d, go through the steps 3-5 (where d is the dimension of $\phi(t)$).

- **Step 3**

$$\beta_j = \beta_{j-1} + f_j g_j$$

$$D(t)_{jj} = \frac{\beta_{j-1} D(t-1)_{jj}}{\beta_j \lambda(t)}$$

$$v_j = g_j$$

$$\mu_j = -\frac{f_j}{\beta_{j-1}}$$

- **Step 4**

For i = 1,...., j-1, do step 5 (for j = 1, skip step 5).

- **Step 5**

Compute

$$U(t)_{ij} = U(t-1)_{ij} + v_i \mu_j$$

$$v_i = v_i + U(t-1)_{ij} v_j$$

- **Step 6**

$$\overline{K}(t) = \begin{bmatrix} v_1 \\ \vdots \\ \vdots \\ v_d \end{bmatrix}, \quad K(t) = \frac{\overline{K}(t)}{\beta_d}$$

- **Step 7**

$$\hat{\theta}(t) = \hat{\theta}(t-1) + K(t)\left[y(t) - \phi(t)\hat{\theta}(t-1) \right]$$

3.1.5 Augmented UDU^T Factorization algorithm

The Augmented UDU^T (AUD) algorithm for system identification proposed by Niu *et al.*(1990) is an extension to the UD algorithm of Bierman (1977). This algorithm can be realized by rearranging the regressor vector and augmenting the covariance matrix of Bierman's UD algorithm. Niu *et al.*(1990) have shown that AUD and UD algorithms are direct extensions of the recursive least squares (RLS) algorithm. In the AUD algorithm it is possible to simultaneously estimate the model parameters and compute the loss functions for all model orders.

In the AUD factorization algorithm the regressor and parameter vectors are defined as follows:

$$\begin{aligned} \phi_r(t) &= [-y(t-r),\ u(t-r),\ \cdots\cdots,\ -y(t-1),\ u(t-1),\ -y(t)]^T \\ &= [h_r(t)\quad -y(t)]^T \end{aligned} \tag{3.36}$$

$$\hat{\theta}_r(t) = [\,a_r,\ b_r,\ \cdots\cdots,\ a_2,\ b_2,\ a_1,\ b_1] \tag{3.37}$$

where r represents the model order. The difference between this regressor and parameter vectors and the one in RLS is that the current measurement of the output is included as the last element in the regressor and the output and input elements appear consecutively as pairs rather than sequentially.

The dimension of $\phi_r(t)$ is $d = 2r + 1$.

Define a new data product matrix as follows:

$$S_r(t) = \left[\sum_{j=1}^{t} \lambda^{t-j} \phi_r(j) \phi_r^T(j)\right] \tag{3.38}$$

with a dimension of (2r+1) x (2r+1).

The new covariance matrix, called the augmented information matrix (AIM) by Niu *et al.*(1990) is defined as an inverse of the data product matrix.

$$C_r(t) \triangleq S_r^{-1}(t)$$

Decomposing the covariance matrix into UDU^T form, yields:

$$C_r(t) = U_r(t) D_r(t) U_r^T(t) \tag{3.39}$$

where, $U_r(t)$ the upper-triangular matrix of dimension (2r+1) x (2r+1) as shown below:

$$U_r(t) = \begin{bmatrix} 1 & \underline{\zeta}_0(t-r) & & & & & & \\ & 1 & \underline{\hat{\theta}}_1(t-r+1) & \underline{\zeta}_1(t-r+1) & & & & \\ & & 1 & & & \underline{\hat{\theta}}_{r-1}(t-1) & \underline{\zeta}_{r-1}(t-1) & \underline{\hat{\theta}}_r(t) \\ & & & 1 & & & & \\ & & & & \ddots & & & \\ & & 0 & & & & 1 & \\ & & & & & & & 1 \end{bmatrix} \tag{3.40}$$

where,

$$\hat{\theta}_r(t) = \left[\sum_{j=1}^{t} \lambda^{t-j} h_r(j) h_r^T(j)\right]^{-1} \sum_{j=1}^{t} \lambda^{t-j} h_r(j) y(j) \tag{3.41}$$

provides the parameter estimates for the r^{th} order model. More generally,

$$\hat{\theta}_{r-i}(t-i) = \left[\sum_{j=1-i}^{t-i} \lambda^{t-i-j} h_{r-i}(j) h_{r-i}^{T}(j) \right]^{-1} \sum_{j=1-i}^{t-i} \lambda^{t-i-j} h_{r-i}(j) y(j) \tag{3.42}$$

provides the parameter estimates for the i^{th} order model, with $i = 1,.....r$.

and, $D_r(t)$ is a diagonal matrix of dimension $(2r+1)$ x $(2r+1)$

$$D_r^{-1}(t) = \begin{bmatrix} J_0(t-r) & & & & & \\ & L_0(t-r) & 0 & & & \\ & & \ddots & & & \\ & & & J_{r-1}(t-1) & & \\ & 0 & & & L_r(t-1) & \\ & & & & & J_r(t) \end{bmatrix} \tag{3.43}$$

where,

$$J_r(t) = \sum_{j=1}^{t} y^2(j) - \hat{\theta}_r^T(t) \left[\sum_{j=1}^{t} \lambda^{t-j} h_r(j) h_r^T(j) \right] \hat{\theta}_r(t) \tag{3.44}$$

is the loss function of the r^{th} order model. More generally,

$$J_{r-i}(t) = \sum_{j=1-i}^{t-i} y^2(j) - \hat{\theta}_{r-i}^T(t-i) \left[\sum_{j=1-i}^{t-i} \lambda^{t-i-j} h_{r-i}(j) h_r^T(j) \right] \hat{\theta}_{r-i}(t-i) \tag{3.45}$$

is the loss function of the i^{th} order model, with $i = 1,....., r$.

$\zeta_{r-i}(t-i)$ and $L_{r-i}(t-i)$ in $U_r(t)$ and $D_r(t)$, for $i = 0,...r$ are intermediate variables, with the forms similar to eqn 3.42 and 3.45 respectively.

For the mathematical details, please refer to Niu *et al.*(1990).

By decomposing $C_r(t)$ into UDU^T form, it is possible to simultaneously obtain the parameter estimates and the loss function from order 1 to r where r is the user-specified

maximum possible order of the real process. If the UD algorithm outlined in the previous section is used with the AIM, then the modified UD algorithm is called the Augmented UD (AUD) algorithm for identification. It simultaneously and recursively identifies the model order and the parameter estimates of the process. The model order can be obtained from the loss function values for different model orders from 0 to r. The model order at which there is a steep descent in the value of the loss function from the previous model order would be the minimal model order for the process. This will be illustrated in greater detail in the next section where the same algorithm will be extended to identify MIMO systems with distinct time-delays.

The AUD algorithm for identification, which is very similar to the UD algorithm discussed in the previous section, is presented in a stepwise manner as follows:

At time t = 0, initialize the algorithm with $C_r(0) = U_r(0)D_r(0)U_r^T(0) = \sigma^2 I$, where σ is a large integer of the order of 100.

At time step t > 0,

construct the data vector according to eqn. 3.36, then update $U_r(t\text{-}1)$ and $D_r(t\text{-}1)$ by performing steps 1-5.

- **Step 1**

$$f = U_r^T(t-1)\phi_r(t)$$

$$g = D_r(t-1)f$$

$$\beta_0 = \lambda(t)$$

- **Step 2** For j = 1,.....,d (d $\triangleq$ 2r + 1) go through steps 3-5.

- **Step 3** Compute

$$\beta_j = \beta_{j-1} + f_j g_j$$

$$D_r(t)_{jj} = \frac{\beta_{j-1} D_r(t-1)_{jj}}{\beta_j \lambda(t)}$$

$$v_j = g_j$$

$$\mu_j = -\frac{f_j}{\beta_{j-1}}$$

- **Step 4** For i = 1,....., j-1, go through step 5 (skip step 5 if j = 1).
- **Step 5**

$$U_r(t)_{ij} = U_r(t-1)_{ij} + v_i \mu_j$$

$$v_i = v_i + U_r(t-1)_{ij} v_j$$

It is worth noting here that this algorithm does not have steps 6 and 7 of the UD algorithm. This is because these steps simply repeat the steps 1-5 for a higher order model. In the AUD algorithm this is done by including the current output data *y(t)* in the data vector (eqn 3.36) and thus increasing the dimension of the variables such as *f, g, U(t), D(t)* by 1. The parameter update vector θ, calculated in step 7 of the UD algorithm, is included in the last column of the U matrix in the AUD algorithm.

Thus, the AUD algorithm is more compact than the Bierman's UD algorithm and it can also simultaneously identify the model order of the process.

3.2 Augmented UD factorization algorithm for identification of MIMO systems with distinct delays

In section 2.2, parametric equivalence between the MIMO system in the TFM form and ARMA form was established. It was proved that a MIMO system with *m-inputs* and *n-outputs* with distinct time-delays can always be represented by the ARX form. The structure for representing a MIMO system so that it is suitable for identification is again reproduced here as:

$$\begin{bmatrix} p_1(z^{-1}) & 0 & \cdots & \cdots & 0 \\ 0 & p_2(z^{-1}) & 0 & \cdots & 0 \\ \vdots & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & p_n(z^{-1}) \end{bmatrix} \begin{bmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ \vdots \\ y_n(t) \end{bmatrix} =$$

$$\begin{bmatrix} q_{11}(z^{-1})z^{-d_{11}} & q_{12}(z^{-1})z^{-d_{12}} & \cdots & \cdots & \cdots & q_{1m}(z^{-1})z^{-d_{1m}} \\ q_{21}(z^{-1})z^{-d21} & \cdots & & \cdots & \cdots & q_{2m}(z^{-d_{2m}} \\ \vdots & \vdots & & \vdots & \vdots & \vdots \\ \vdots & \vdots & & \vdots & \vdots & \vdots \\ q_{n1}(z^{-1})z^{-d_{n1}} & \cdots & & \cdots & \cdots & q_{nm}(z^{-1})z^{-d_{nm}} \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ \vdots \\ u_m(t) \end{bmatrix} \tag{3.46}$$

In many recent studies in the identification of a MIMO system, the scheme to identify all the MIMO model parameters in a single parameter matrix has been used (Niu *et al.*,1990). This can be computationally very heavy and data management can be difficult. Also it may not be possible to complete the identification of all parameters within the sampling interval and make these estimates available for the controller calculations.

Moreover, if there are distinct delays in the system, which exist naturally in many chemical systems, there will be many zeros corresponding to the values of the delays in the model. Thus, the already difficult problem of identifying parameters of a MIMO system can become even more difficult to handle, since there may be some extra parameters which are basically "zeros" to represent delay, that have to be identified by the algorithm. An identification algorithm minimizes the cost function of the prediction errors with respect to the parameters of the model. Thus, with some extra parameters in the form of zeros, the algorithm will have extra degrees of freedom to minimize the cost function, and it may under some circumstances, assign non-zero values to the parameters which are known to be zeros. This will corrupt the estimates of the actual process parameters.

Many such problems can be resolved by using the idea of decomposing a MIMO system into as many MISO (Multi-input Single Output) subsystems as the number of outputs in the system. This will break the huge multi-dimensional identification problem of estimating all elements of a transfer function matrix of a MIMO system into a number of small single-dimensional problems that can be solved in parallel. Also, distinct time-delays can be handled without having unnecessary zeros in the parameter vector, and thus a minimal number of parameters are identified. The MIMO system with *m-inputs* and *n-outputs* in eqn.3.46 can be decomposed into *n* MISO subsystems as shown below:

$$p_1(z^{-1})y_1(t) = \begin{bmatrix} q_{11}(z^{-1}) & q_{12}(z^{-1}) & \cdots & \cdots & q_{1m}(z^{-1}) \end{bmatrix} \begin{bmatrix} u_1(t-d_{11}) \\ u_2(t-d_{12}) \\ \vdots \\ \vdots \\ u_m(t-d_{1m}) \end{bmatrix} \tag{3.47}$$

$$p_2(z^{-1})y_2(t) = \begin{bmatrix} q_{21}(z^{-1}) & q_{22}(z^{-1}) & \cdots & \cdots & q_{2m}(z^{-1}) \end{bmatrix} \begin{bmatrix} u_1(t-d_{21}) \\ u_2(t-d_{22}) \\ \vdots \\ \vdots \\ u_m(t-d_{2m}) \end{bmatrix} \tag{3.48}$$

$$\vdots \qquad \vdots \qquad \vdots \qquad \vdots$$

$$\vdots \qquad \vdots \qquad \vdots \qquad \vdots$$

$$p_n(z^{-1})y_n(t) = \begin{bmatrix} q_{n1}(z^{-1}) & q_{n2}(z^{-1}) & \cdots & \cdots & q_{nm}(z^{-1}) \end{bmatrix} \begin{bmatrix} u_1(t-d_{n1}) \\ u_2(t-d_{n2}) \\ \vdots \\ \vdots \\ u_m(t-d_{nm}) \end{bmatrix} \tag{3.49}$$

where,

$$p_i(z^{-1}) = 1 + p_{i1}z^{-1} + p_{i2}z^{-2} + \cdots\cdots + p_{ir_i}z^{-r_i} \tag{3.50}$$

i = 1,2,.....n (number of outputs), r_i is the degree of the p_i polynomial and

$$q_{ij}(z^{-1}) = q_{ij,1}z^{-1} + q_{ij,2}z^{-2} + \cdots\cdots + q_{ij,r_i}z^{-r_i} \tag{3.51}$$

j = 1,2,.....m (number of inputs), and r_i is the degree of the q_{ij} polynomial.

For each of these MISO subsystems, the augmented data vector or the regressor vector and the parameter vector can be written as follows:

For MISO 1,

$$\phi_1(t) = \left[-y_1(t-r_1),\ u_1(t-d_{11}-r_1),\ \cdots,\ u_m(t-d_{1m}-r_1),\ \cdots\cdots \right.$$
$$\left. \cdots\cdots,\ -y_1(t-1),\ u_1(t-d_{11}-1),\ \cdots,\ u_m(t-d_{1m}-1),\ -y_1(t)\right]^T \qquad (3.52)$$

$$\hat{\theta}_1 = \left[p_{1r_1},\ q_{11,r_1},\ \cdots,\ q_{1m,r_1},\ \cdots\cdots\cdots,\ p_{11},\ q_{11,1},\ \cdots,\ q_{1m,1}\right]$$

length of $\phi_1(t) = d_1 =$ (no. of inputs + 1)*r_1 + 1

More generally, for an i^{th} MISO subsystem,

$$\phi_i(t) = \left[-y_i(t-r_i),\ u_1(t-d_{i1}-r_i),\ \cdots,\ u_m(t-d_{im}-r_i),\ \cdots\cdots \right.$$
$$\left. \cdots\cdots,\ -y_i(t-1),\ u_1(t-d_{i1}-1),\ \cdots,\ u_m(t-d_{im}-1),\ -y_i(t)\right]^T$$

$$\phi_i(t) = \left[\ h_i(t),\ -y_i(t)\ \right]^T \qquad (3.53)$$

$$\hat{\theta}_i = \left[p_{ir_i},\ q_{i1,r_i},\ \cdots,\ q_{im,r_i},\ \cdots\cdots\cdots,\ p_{i1},\ q_{i1,1},\cdots q_{i1,1},\ \cdots,\ q_{im,1}\right]$$

length of $\phi_i(t) = d_i =$ (no. of inputs + 1)*r_i + 1 $\quad \{ i = 1, 2, \ldots\ldots, n \}$

Define a covariance matrices for each MISO subsystem as in eqn.3.38.

$$C_i(t) = \left[\sum_{k=1}^{t} \lambda^{t-k} \phi_i(k) \phi_i^T(k) \right]^{-1} \qquad (3.54)$$

i = 1, 2, ..., n

Decompose each covariance matrix into UDU^T form,

$$C_i(t) = U_i(t)\,D_i(t)\,U_i^T(t) \tag{3.55}$$

with, $U_i(t)$ the upper-triangular matrix of dimension d_i x d_i is shown below:

$$U_i(t) = \begin{bmatrix} 1 & \underline{\alpha}_0(t-r_i) & & & & & & \\ & 1 & \underline{\hat{\theta}}_1(t-r_i+1) & \cdots & \underline{\alpha}_1(t-r_i+1) & & & \\ & & 1 & & & \underline{\hat{\theta}}_{r_i-1}(t-1) & \cdots & \underline{\hat{\alpha}}_{r_i-1}(t-1) & \underline{\hat{\theta}}_{r_i}(t) \\ & & & 1 & & & & & \\ & & & & \ddots & & & & \\ & & & & & 1 & & & \\ & & & & & & \ddots & & \\ & & 0 & & & & & 1 & \\ & & & & & & & & 1 \end{bmatrix} \tag{3.56}$$

where,

$$\hat{\theta}_{r_i}(t) = \left[\sum_{j=1}^{t} \lambda^{t-j} h_{r_i}(j)\, h_{r_i}^T(j) \right]^{-1} \sum_{j=1}^{t} \lambda^{t-j} h_{r_i}(j)\, y(j) \tag{3.57}$$

provides the parameter estimates for r_i^{th} order model of the i^{th} MISO subsystem. More generally,

$$\hat{\theta}_{r_i-k}(t-k) = \left[\sum_{j=1}^{t-k} \lambda^{t-k-j} h_{r_i-k}(j)\, h_{r_i-k}^T(j) \right]^{-1} \sum_{j=1}^{t-k} \lambda^{t-k-j} h_{r_i-k}(j)\, y(j) \tag{3.58}$$

provides the parameter estimates for the k^{th} order model, with $k = 1,\ldots\ldots,r_i$.

and $D_i(t)$ is a diagonal matrix of dimension d_i x d_i,

$$D_i^{-1}(t) = \begin{bmatrix} J_0(t-r_i) & & & & & & \\ & \ddots & & & & & \\ & & L_0(t-r_i) & 0 & & & \\ & & & \ddots & & & \\ & & & & J_{r_i-1}(t-1) & & \\ & & & & & \ddots & \\ & 0 & & & & L_{r_i-1}(t-1) & \\ & & & & & & J_{r_i}(t) \end{bmatrix} \tag{3.59}$$

where,

$$J_{r_i}(t) = \sum_{j=1}^{t} y^2(j) - \hat{\theta}_{r_i}^T(t)\left[\sum_{j=1}^{t} \lambda^{t-j} h_{r_i}(j) h_{r_i}^T(j)\right]\hat{\theta}_{r_i}(t) \tag{3.60}$$

is the loss function of the r_i^{th} order model in i^{th} MISO subsystem. More generally,

$$J_{r_i-k}(t) = \sum_{j=1}^{t-k} y^2(j) - \hat{\theta}_{r_i-k}^T(t)\left[\sum_{j=1}^{t-k} \lambda^{t-j} h_{r_i-k}(j) h_{r_i-k}^T(j)\right]\hat{\theta}_{r_i-k}(t) \tag{3.61}$$

is the loss function of the k^{th} order model, with $k = 1,\ldots\ldots,r_i$.

The AUDI algorithm discussed in sec 3.1.5 is used here for estimation of parameters of each MISO subsystem. It will recursively estimate the model parameters of each MISO subsystem, and the loss function values from order 1 to r_i, where r_i is the order of the i^{th} MISO subsystem. From the values of loss functions corresponding to each model order, it is possible to estimate the optimal model order for every MISO subsystem.

The model order is obtained as follows:

For an i^{th} MISO subsystem, find the values the loss function values as discussed above. These are the J_i values in the $D_i^{-1}(t)$ shown in eqn 3.59. These values correspond to the loss function of the i^{th} MISO subsystem for order from 0 to r_i. Plotting these J_i values at each model order. The plot would be of the following type.

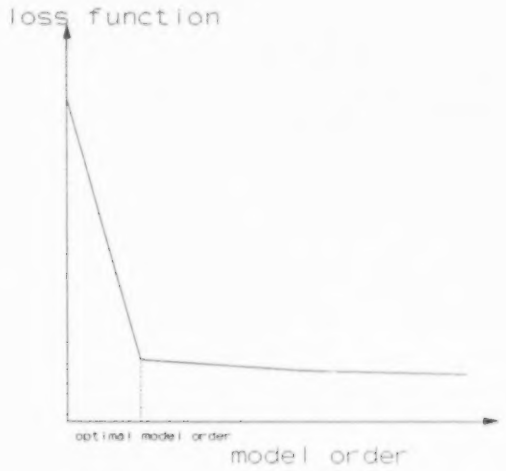


Figure 3.1 A diagram to show the optimal model order of a system

The model order, at which there is the steepest drop in the value of the loss function, is the optimal model order. This is because at the model order one less than the optimal one, the cost function is significantly large and for the model orders higher than the optimal model order, the extra parameters in the model may not cause a significant change in the value of the cost function.

3.2.1 Openloop study of the MIMO AUDI algorithm

The MIMO AUDI algorithm has been evaluated by application to a 2 x 2 system. The input-output data were obtained from a pilot-scale continuous-stirred tank heater. The equipment, hardware and the software for implementation is described elsewhere in this thesis (please see chapter 5).

The two inputs to the process are cold water % valve opening (C_w) and hot water % valve opening (H_w). The two outputs from the process are temperature (T in °C) of the exit water and level of the water (L in % of the height of the tank) in the tank. The excitation in the form of step type inputs is given to the process through the two inputs. The outputs, i.e. the temperature (T) and the level (L) are measured at every sample instant. The sampling interval used is 4 seconds. The data-acquisition is done simultaneously and all the measurements of input and outputs are stored.

The structure of this 2 inputs, 2 outputs system is the same as that discussed in chapter 2 (eqn 2.46). The structure is as shown:

$$\begin{bmatrix} p_1(z^{-1}) & 0 \\ 0 & p_2(z^{-1}) \end{bmatrix} \begin{bmatrix} T(t) \\ L(t) \end{bmatrix} = \begin{bmatrix} q_{11}(z^{-1})z^{-d_{11}} & q_{12}(z^{-1})z^{-d_{12}} \\ q_{21}(z^{-1})z^{-d_{21}} & q_{22}(z^{-1})z^{-d_{22}} \end{bmatrix} \begin{bmatrix} C_w(t) \\ H_w(t) \end{bmatrix} + \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} \qquad (3.62)$$

where, $p_1(z^{-1})$, $p_2(z^{-1})$, $q_{11}(z^{-1})$, $q_{12}(z^{-1})$, $q_{21}(z^{-1})$, $q_{22}(z^{-1})$ are the polynomials in terms of the backward shift operator z^{-1}.

d_{11}, d_{12}, d_{21}, d_{22} are the delays in the model, which need to be determined in advance. One method to estimate the delays are by off-line step response studies of the process. For this system, the values of the time-delay at the nominal operating condition was found to be 4 seconds for each element of the Q matrix. Since a sampling time of 4

seconds is used, the time-delays in discrete domain will be $d_{11} = 1$; $d_{12} = 1$; $d_{21} = 1$; $d_{22} = 1$.

The error of closure $\underline{d}$ is:

$$\begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = \begin{bmatrix} p_1(z^{-1}) & 0 \\ 0 & p_2(z^{-1}) \end{bmatrix} \begin{bmatrix} T_s \\ L_s \end{bmatrix} - \begin{bmatrix} q_{11}(z^{-1})z^{-d_{11}} & q_{12}(z^{-1})z^{-d_{12}} \\ q_{21}(z^{-1})z^{-d_{21}} & q_{22}(z^{-1})z^{-d_{22}} \end{bmatrix} \begin{bmatrix} C_{w_s} \\ H_{w_s} \end{bmatrix} \tag{3.63}$$

where, T_s, L_s, C_{ws}, H_{ws} are the steady-state values of the variables.

The MIMO AUD algorithm for *m-inputs* and *n-outputs* was discussed in the previous section. The same algorithm will be used for a specific 2 x 2 MIMO system. The system described in eqn 3.62 is decomposed into 2 MISO subsystems as follows:

$$\begin{aligned} p_1(z^{-1})T(t) &= \begin{bmatrix} q_{11}(z^{-1}) & q_{12}(z^{-1}) \end{bmatrix} \begin{bmatrix} C_w(t-d_{11}) \\ H_w(t-d_{12}) \end{bmatrix} + d_1 \\ p_2(z^{-1})L(t) &= \begin{bmatrix} q_{21}(z^{-1}) & q_{22}(z^{-1}) \end{bmatrix} \begin{bmatrix} C_w(t-d_{21}) \\ H_w(t-d_{22}) \end{bmatrix} + d_2 \end{aligned} \tag{3.64}$$

where,

$$\begin{aligned} p_i(z^{-1}) &= 1 + p_{i1}z^{-1} + \ldots\ldots + p_{ir_i}z^{-r_i} \\ q_{ij}(z^{-1}) &= q_{ij,1}z^{-1} + q_{ij,2}z^{-2} + \ldots\ldots + q_{ij,r_i}z^{-r_i} \end{aligned} \tag{3.65}$$

for i,j = 1 to 2.

r_1 = maximum possible order of the model in MISO 1.

r_2 = maximum possible order of the model in MISO 2.

The *p's* and *q's* are the parameters of the model which describes the characteristics of the system, which is to be estimated.

For each of the MISO subsystem, the augmented data vector or the regressor vector and the parameter vector are written as follows:

For MISO subsystem 1

$$\phi_{r_1}(t) = \left[-T(t-r_1),\ C_w(t-d_{11}-r_1),\ H_w(t-d_{12}-r_1),\ \cdots\cdots,\right.$$
$$\left.\cdots\cdots,\ -T(t-1),\ C_w(t-d_{11}-1),\ H_w(t-d_{12}-1),\ 1,\ -T(t)\right]^T \qquad (3.66)$$

$$\hat{\theta}_{r_1} = \left[p_{1r_1},\ q_{11,r_1},\ q_{12,r_1},\ \cdots\cdots\cdots,\ p_{11},\ q_{11,1},\ q_{12,1},\ d_1\right]^T$$

length of $\phi_{r1}(t) = d_1 = (2+1)r_1 + 1 + 1 = 3r_1 + 2$

For MISO subsystem 2

$$\phi_{r_2}(t) = \left[-L(t-r_2),\ C_w(t-d_{21}-r_2),\ H_w(t-d_{22}-r_2),\ \cdots\cdots,\right.$$
$$\left.\cdots\cdots,\ -L(t-1),\ C_w(t-d_{21}-1),\ H_w(t-d_{22}-1),\ 1,\ -L(t)\right]^T \qquad (3.67)$$

$$\hat{\theta}_{r_2} = \left[p_{2r_2},\ q_{21,r_2},\ q_{22,r_2},\ \cdots\cdots\cdots,\ p_{21},\ q_{21,1},\ q_{22,1},\ d_2\right]^T$$

length of $\phi_{r2}(t) = d_2 = (2+1)r_2 + 1 + 1 = 3r_2 + 2$

The covariance matrix for both MISO subsystems are,

$$C_1(t) = \left[\sum_{k=1}^{t} \lambda_1^{t-k}\phi_1(k)\phi_1^T(k)\right]^{-1} \qquad C_2(t) = \left[\sum_{k=1}^{t} \lambda_2^{t-k}\phi_2(k)\phi_2^T(k)\right]^{-1} \qquad (3.68)$$

λ_1 and λ_2 are the forgetting factors for each MISO subsystem.

Decomposing the covariance matrices into UDU^T form yields:

$$C_1(t) = U_1(t)D_1(t)U_1^T(t) \qquad C_2(t) = U_2(t)D_2(t)U_2^T(t) \qquad (3.69)$$

where, $U_1(t)$ and $U_2(t)$ are the upper-triangular matrices and $D_1(t)$ and $D_2(t)$ are the

diagonal matrices for each MISO subsystem respectively.

These matrices will have the same structure as shown in eqn 3.56 and eqn 3.59.

The parameter vectors at each time instant for both MISO subsystems can be obtained as follows:

θ_{r1} = d_1^{th} column of the $U_1(t)$ matrix.

θ_{r2} = d_2^{th} column of the $U_2(t)$ matrix.

The D matrices will be as follows:

$D_1^{-1}(t) = \text{diag}[\, J_0(t\text{-}r_1), \bar{L}_0(t\text{-}r_1), L_0(t\text{-}r_1), \cdots, \cdots, J_{r1\text{-}1}(t\text{-}1), \bar{L}_{r1\text{-}1}(t\text{-}1), L_{r1\text{-}1}(t\text{-}1), R_1(t\text{-}1), J_{r1}(t)]$

$D_2^{-1}(t) = \text{diag}[\, J_0(t\text{-}r_2), \bar{L}_0(t\text{-}r_2), L_0(t\text{-}r_2), \cdots, \cdots, J_{r2\text{-}1}(t\text{-}1), \bar{L}_{r2\text{-}1}(t\text{-}1), L_{r2\text{-}1}(t\text{-}1), R_2(t\text{-}1), J_{r2}(t)]$

The J's in the above equations are the loss functions corresponding to the model order shown in their subscripts. L's and R's are intermediate variables.

3.2.1.1 Identification of the order of the process

The input-output data from the process were processed using the algorithm discussed above to identify the order of each MISO subsystem, one for temperature (T) and the other for level (L).

The input-output data from the process is shown in figures 3.2 and 3.3.

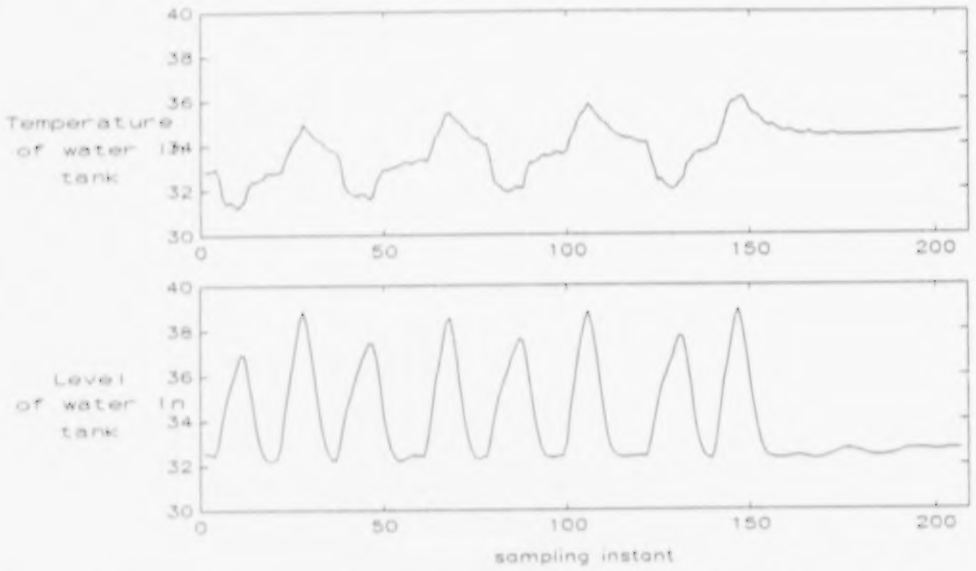


Figure 3.2 Output data from the continuous stirred tank heater

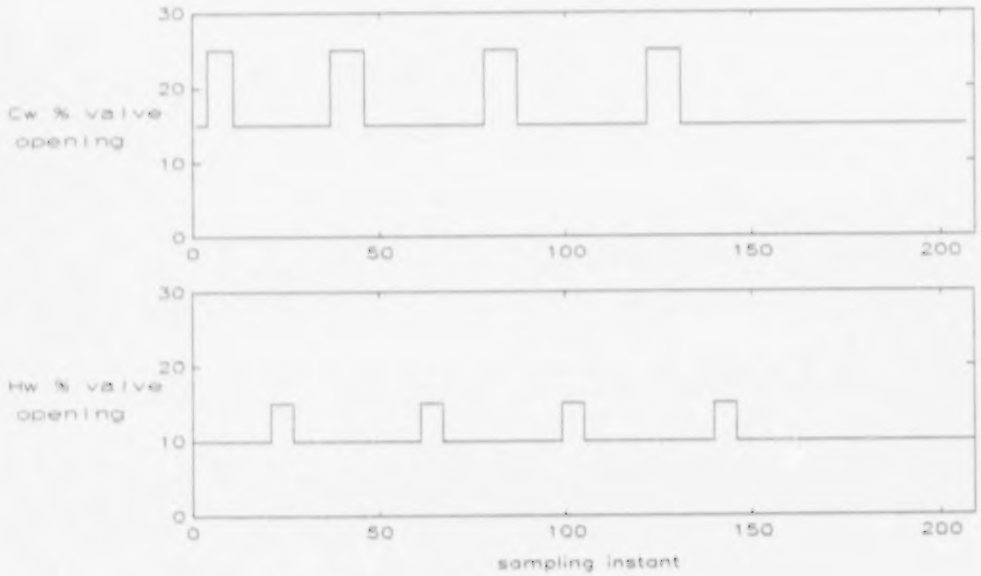


Figure 3.3 Input data from continuous stirred tank heater

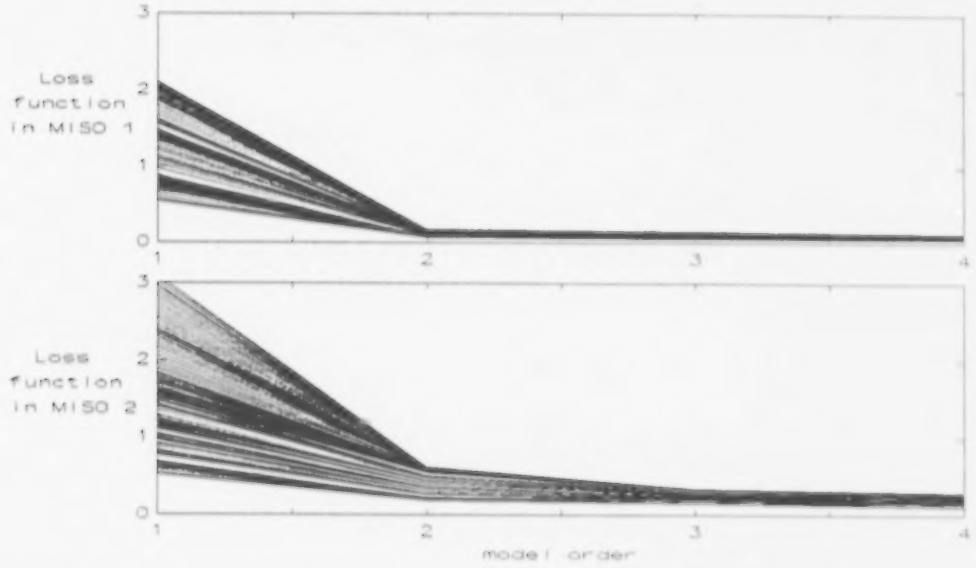


Figure 3.4 Loss functions for MISO subsystems corresponding to temperature and level loops respectively of the CSTH.

The maximum possible order for MISO 1 was taken to be 4 and that for the MISO 2 was also taken to be 4. The identification algorithm was used with a constant forgetting factor for both MISO subsystems set at $\lambda_1 = 1$ and $\lambda_2 = 1$.

The initial values of the covariance matrices were,

$C_1(0) = 100$ and $C_2(0) = 100$.

The loss functions for both MISO subsystem were obtained at each sampling instant and are shown in figure 3.4.

It can be concluded from figure 3.4, that the optimal model order for MISO subsystem 1 (temperature loop) is 2 and that for MISO subsystem 2 (level loop) is also 2.

Thus, the length of the data vector in MISO 1 = (2 + 1)*2 + 1 + 1 = 8

and the length of the data vector in MISO 2 = (2 + 1)*2 + 1 + 1 = 8

3.2.1.2 Identification of ARMAX and ARIMAX based models with no forgetting of past data

In the previous subsection, it was shown that each MISO subsystem is second order. In this subsection, the parameters for two second order MISO models of the process are estimated based on ARMAX and ARIMAX representations.

ARMAX model for the 2 x 2 MIMO system

The 2nd order ARMAX model for the CSTH system can be represented as follows:

$$\begin{bmatrix} 1+p_{11}z^{-1}+p_{12}z^{-2} & 0 \\ 0 & 1+p_{21}z^{-1}+p_{22}z^{-2} \end{bmatrix} \begin{bmatrix} T(t) \\ L(t) \end{bmatrix} = \tag{3.70}$$

$$\begin{bmatrix} q_{11}z^{-d_{11}-1}+q_{12}z^{-d_{11}-2} & q_{13}z^{-d_{12}-1}+q_{14}z^{-d_{12}-2} \\ q_{21}z^{-d_{21}-1}+q_{22}z^{-d_{21}-2} & q_{23}z^{-d_{22}-1}+q_{24}z^{-d_{22}-2} \end{bmatrix} \begin{bmatrix} C_w(t) \\ H_w(t) \end{bmatrix} + \begin{bmatrix} 1+c_1z^{-1} & 0 \\ 0 & 1+c_2z^{-1} \end{bmatrix} \begin{bmatrix} e_1(t) \\ e_2(t) \end{bmatrix} + \begin{bmatrix} d_1 \\ d_2 \end{bmatrix}$$

Since e(t) is not measurable, estimation of the C-polynomial can be done by using the prediction error as proxy terms for e(t). However this will result in poor convergence of the 'c' parameters (Ljung and Soderstrom, 1983). The solution to this problem is to use a design polynomial or a disturbance tailoring filter ($C_e(z^{-1})$), where $1/C_e$ is a low-pass filter (Mohtadi, 1986). It is assumed that $C_e = C$. For this work the value of C_e was taken as $[1 - 0.8z^{-1}]$ for both the MISO subsystems as recommended by McIntosh (1988). The polynomial matrix $C_e(z^{-1})$ then becomes:

$$C_e(z^{-1})^{-1} = \begin{bmatrix} 1-0.8z^{-1} & 0 \\ 0 & 1-0.8z^{-1} \end{bmatrix}^{-1} = \begin{bmatrix} 1+c_1z^{-1} & 0 \\ 0 & 1+c_2z^{-1} \end{bmatrix}^{-1} \quad (3.71)$$

Pre-multiplying both sides of eqn 3.70 by C_e^{-1} gives:

$$\begin{bmatrix} 1+p_{11}z^{-1}+p_{12}z^{-2} & 0 \\ 0 & 1+p_{21}z^{-1}+p_{22}z^{-2} \end{bmatrix} \begin{bmatrix} T_f(t) \\ L_f(t) \end{bmatrix} =$$

$$\begin{bmatrix} q_{11}z^{-d_{11}-1}+q_{12}z^{-d_{11}-2} & q_{13}z^{-d_{12}-1}+q_{14}z^{-d_{12}-2} \\ q_{21}z^{-d_{21}-1}+q_{22}z^{-d_{21}-2} & q_{23}z^{-d_{22}-1}+q_{24}z^{-d_{22}-2} \end{bmatrix} \begin{bmatrix} C_{wf}(t) \\ H_{wf}(t) \end{bmatrix} + \begin{bmatrix} d_{1_f} \\ d_{2_f} \end{bmatrix} \quad (3.72)$$

where, the subscript 'f' denotes filtered values.

For example, the filtered output 1, $T_f(t)$, is given be:

$$T_f(t) = 0.8T_f(t-1) + T(t) \quad (3.73)$$

Using this equation the data vector for MISO 1 and MISO 2 can be constructed as discussed earlier in section 3.2.1. In constructing the data vector, the filtered input-output data are used. The MIMO AUD algorithm discussed in section 3.1.1 is used for each MISO subsystem to identify the parameters of eqn 3.73. The forgetting factor ,λ_i, for each MISO subsystem is kept constant, $\lambda_1(t) = \lambda_2(t) = 1.0$, i.e. there is no forgetting of past data, since the asymptotic memory length used is given by $1/(1-\lambda)$, which in this case will be infinitely long.

The initial value for the covariance matrices were set to, $C_1(0) = 100*I$ and $C_2(0) = 100*I$. To validate the parameter estimates, the actual measured outputs of the process are compared with the estimates outputs obtained from the estimated parameters of the model at every time step. The actual versus estimated outputs and the

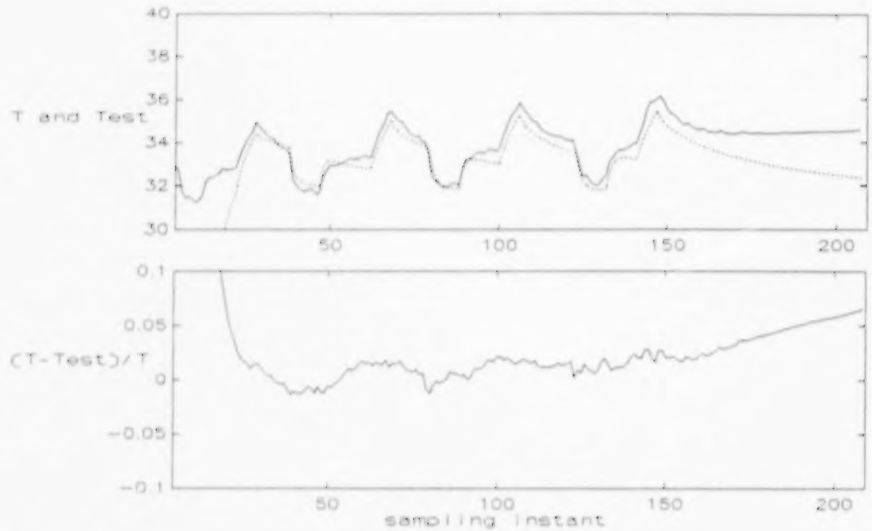


Figure 3.5 Actual and estimated temperature from ARMAX model identification without forgetting factor

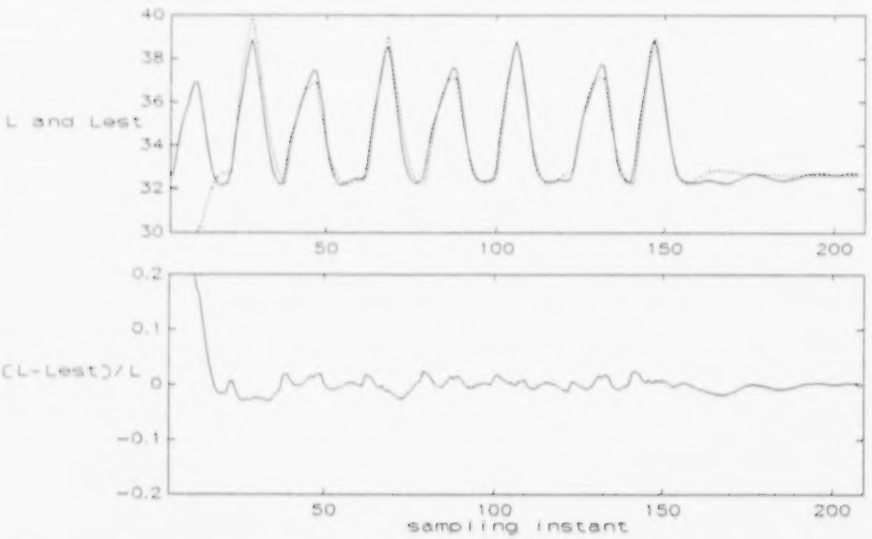


Figure 3.6 Actual and estimated level from ARMAX model identification without forgetting factor

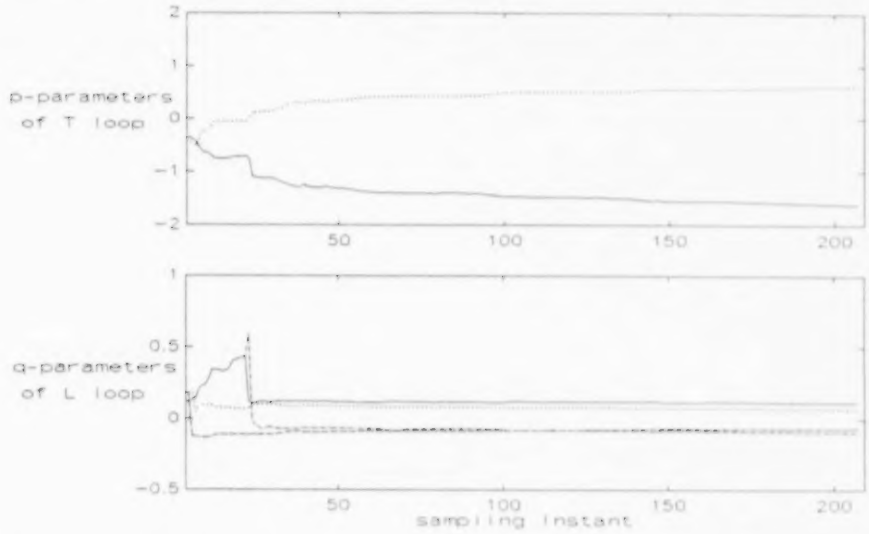


Figure 3.7 Parameter estimates of the MISO subsystem corresponding to the temperature loop in ARMAX representation without forgetting factor

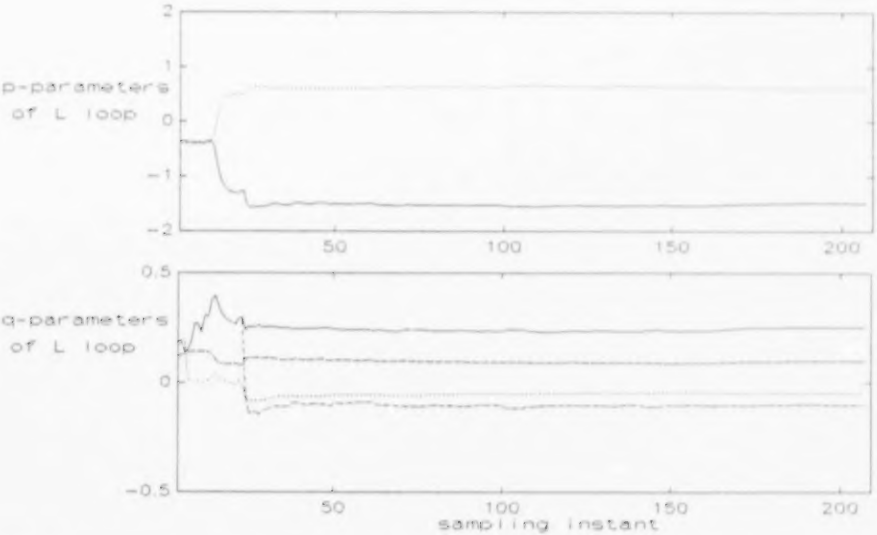


Figure 3.8 Parameter estimates of the MISO subsystem corresponding to the level loop in ARMAX representation without forgetting factor

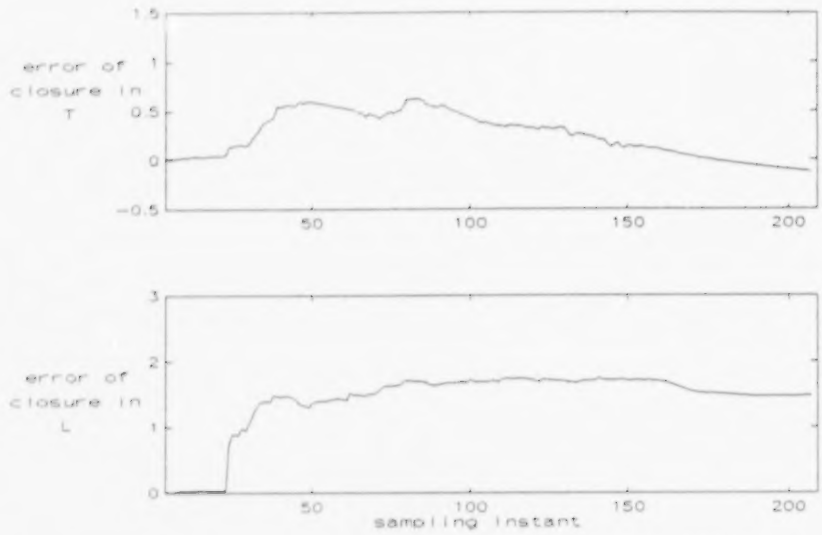


Figure 3.9 Estimates of the error of closure in temperature in ARMAX representation without a forgetting factor

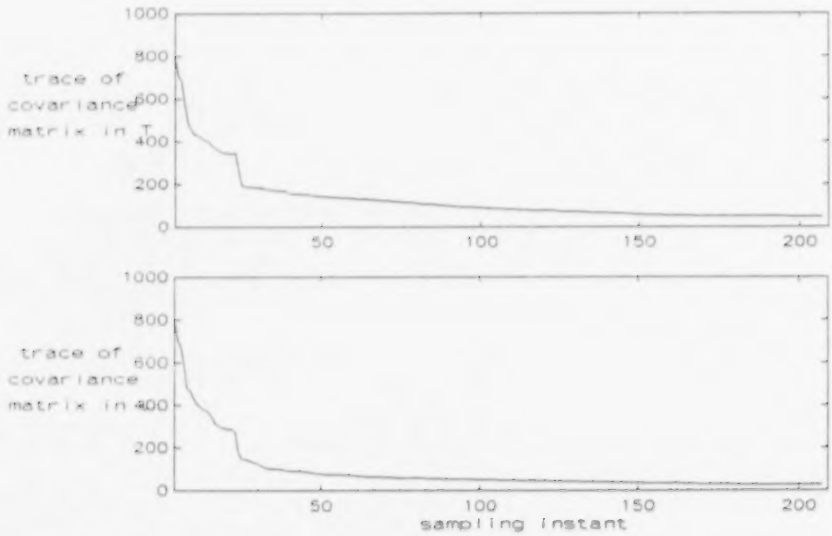


Figure 3.10 Trace of the covariance matrix of the data vector in both MISO subsystems in ARMAX representation without forgetting factor

normalized prediction errors are shown in figures 3.5 and 3.6. The parameter trajectories of both MISO subsystems, are presented in figures 3.7 and 3.8. Figure 3.9 shows the variation of the error of closure or the value of the d_r parameters and figure 3.10 shows the variation of the trace of the covariance matrix. Figure 3.5 shows that after sampling instant # 150 when the input excitation is stopped, the estimated output is unable to track the actual output. With little new information about the process after the 150th sampling instance, the algorithm is not alert enough to track the time-varying process parameters. Another indication of this can also be seen from figure 3.10 where the trace of the covariance matrix of the data vector reduces rapidly and remains at a small value throughout the course of the run. To alleviate this difficulty, various modifications can be incorporated into the identification algorithm. These modifications are discussed in sections 3.2.1.3 and 3.2.1.4 of this chapter.

ARIMAX model identification for the MIMO (2 x 2) system

The 2nd ARIMAX model of the CSTH system can be represented as follows:

$$\begin{bmatrix} 1+p_{11}z^{-1}+p_{12}z^{-2} & 0 \\ 0 & 1+p_{21}z^{-1}+p_{22}z^{-2} \end{bmatrix} \begin{bmatrix} T(t) \\ L(t) \end{bmatrix} =$$

$$\begin{bmatrix} q_{11}z^{-d_{11}-1}+q_{12}z^{-d_{11}-2} & q_{13}z^{-d_{12}-1}+q_{14}z^{-d_{12}-2} \\ q_{21}z^{-d_{21}-1}+q_{22}z^{-d_{21}-2} & q_{23}z^{-d_{22}-1}+q_{24}z^{-d_{22}-2} \end{bmatrix} \begin{bmatrix} C_w(t) \\ H_w(t) \end{bmatrix} + \qquad (3.74)$$

$$\begin{bmatrix} \Delta & 0 \\ 0 & \Delta \end{bmatrix}^{-1} \begin{bmatrix} 1+c_1z^{-1} & 0 \\ 0 & 1+c_2z^{-1} \end{bmatrix} \begin{bmatrix} e_1(t) \\ e_2(t) \end{bmatrix} + \begin{bmatrix} d_1(t) \\ d_2(t) \end{bmatrix}$$

where, $\Delta = 1 - z^{-1}$

Eqn 3.74 can be rearranged to obtain the following form,

$$\begin{bmatrix}\Delta & 0\\ 0 & \Delta\end{bmatrix}\begin{bmatrix}1+p_{11}z^{-1}+p_{12}z^{-2} & 0\\ 0 & 1+p_{21}z^{-1}+p_{22}z^{-2}\end{bmatrix}\begin{bmatrix}T(t)\\ L(t)\end{bmatrix} =$$

$$\begin{bmatrix}\Delta & 0\\ 0 & \Delta\end{bmatrix}\begin{bmatrix}q_{11}z^{-d_{11}-1}+q_{12}z^{-d_{11}-2} & q_{13}z^{-d_{12}-1}+q_{14}z^{-d_{12}-2}\\ q_{21}z^{-d_{21}-1}+q_{22}z^{-d_{21}-2} & q_{23}z^{-d_{22}-1}+q_{24}z^{-d_{22}-2}\end{bmatrix}\begin{bmatrix}C_w(t)\\ H_w(t)\end{bmatrix}+\begin{bmatrix}1+c_1z^{-1} & 0\\ 0 & 1+c_2z^{-1}\end{bmatrix}\begin{bmatrix}e_1\\ e_2\end{bmatrix} \quad (3.75)$$

or

$$\begin{bmatrix}1+p'_{11}z^{-1}+p'_{12}z^{-2}+p'_{13}z^{-3} & 0\\ 0 & 1+p'_{21}z^{-1}+p'_{22}z^{-2}+p'_{23}z^{-3}\end{bmatrix}\begin{bmatrix}T(t)\\ L(t)\end{bmatrix} =$$

$$\begin{bmatrix}q_{11}z^{-d_{11}-1}+q_{12}z^{-d_{11}-2} & q_{13}z^{-d_{12}-1}+q_{14}z^{-d_{12}-2}\\ q_{21}z^{-d_{21}-1}+q_{22}z^{-d_{21}-2} & q_{23}z^{-d_{22}-1}+q_{24}z^{-d_{22}-2}\end{bmatrix}\begin{bmatrix}\Delta C_w(t)\\ \Delta H_w(t)\end{bmatrix}+\begin{bmatrix}1+c_1z^{-1} & 0\\ 0 & 1+c_2z^{-1}\end{bmatrix}\begin{bmatrix}e_1(t)\\ e_2(t)\end{bmatrix} \quad (3.76)$$

In this formulation the *d* term is cancelled due to the differencing operation using Δ. However in eqn 3.76 the degree of the *p-polynomials* has increased by 1 due to the addition of the Δ operator. Filtered values of the outputs and the incremental inputs are used to construct the data vectors for both MISO subsystems. The C_e filter equation is the same as in eqn 3.71. The model in the ARIMAX form with filtered incremental inputs and outputs is follows:

$$\begin{bmatrix}1+p'_{11}z^{-1}+p'_{12}z^{-2}+p'_{13}z^{-3} & 0\\ 0 & 1+p'_{21}z^{-1}+p'_{22}z^{-2}+p'_{23}z^{-3}\end{bmatrix}\begin{bmatrix}T_f(t)\\ L_f(t)\end{bmatrix} =$$

$$\begin{bmatrix}q_{11}z^{-d_{11}-1}+q_{12}z^{-d_{11}-2} & q_{13}z^{-d_{12}-1}+q_{14}z^{-d_{12}-2}\\ q_{21}z^{-d_{21}-1}+q_{22}z^{-d_{21}-2} & q_{23}z^{-d_{22}-1}+q_{24}z^{-d_{22}-2}\end{bmatrix}\begin{bmatrix}\Delta C_{w_f}(t)\\ \Delta H_{w_f}(t)\end{bmatrix}+\begin{bmatrix}e_1(t)\\ e_2(t)\end{bmatrix} \quad (5.25)$$

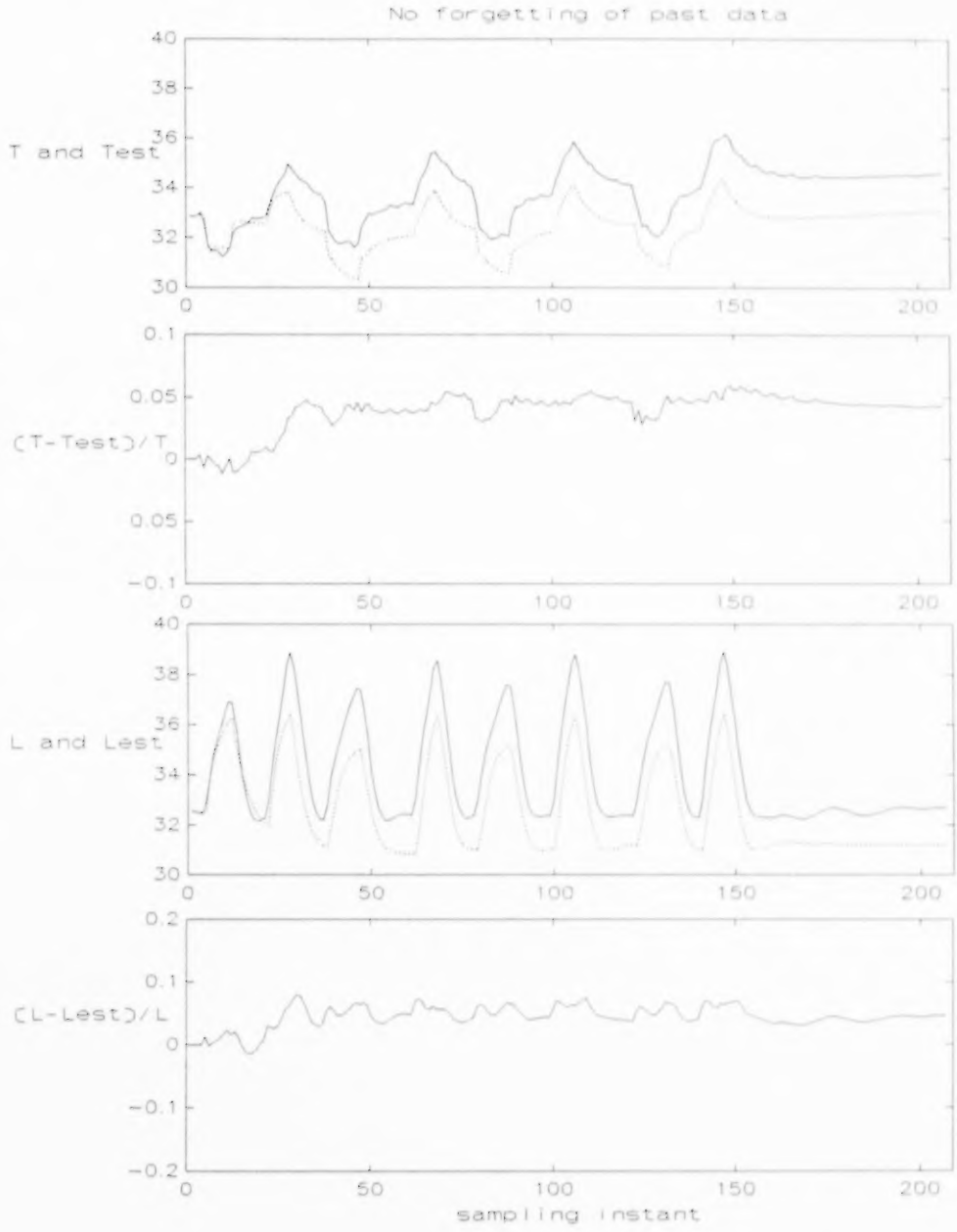


Figure 3.11 Actual and estimated temperature and level from ARIMAX model identification without forgetting factor

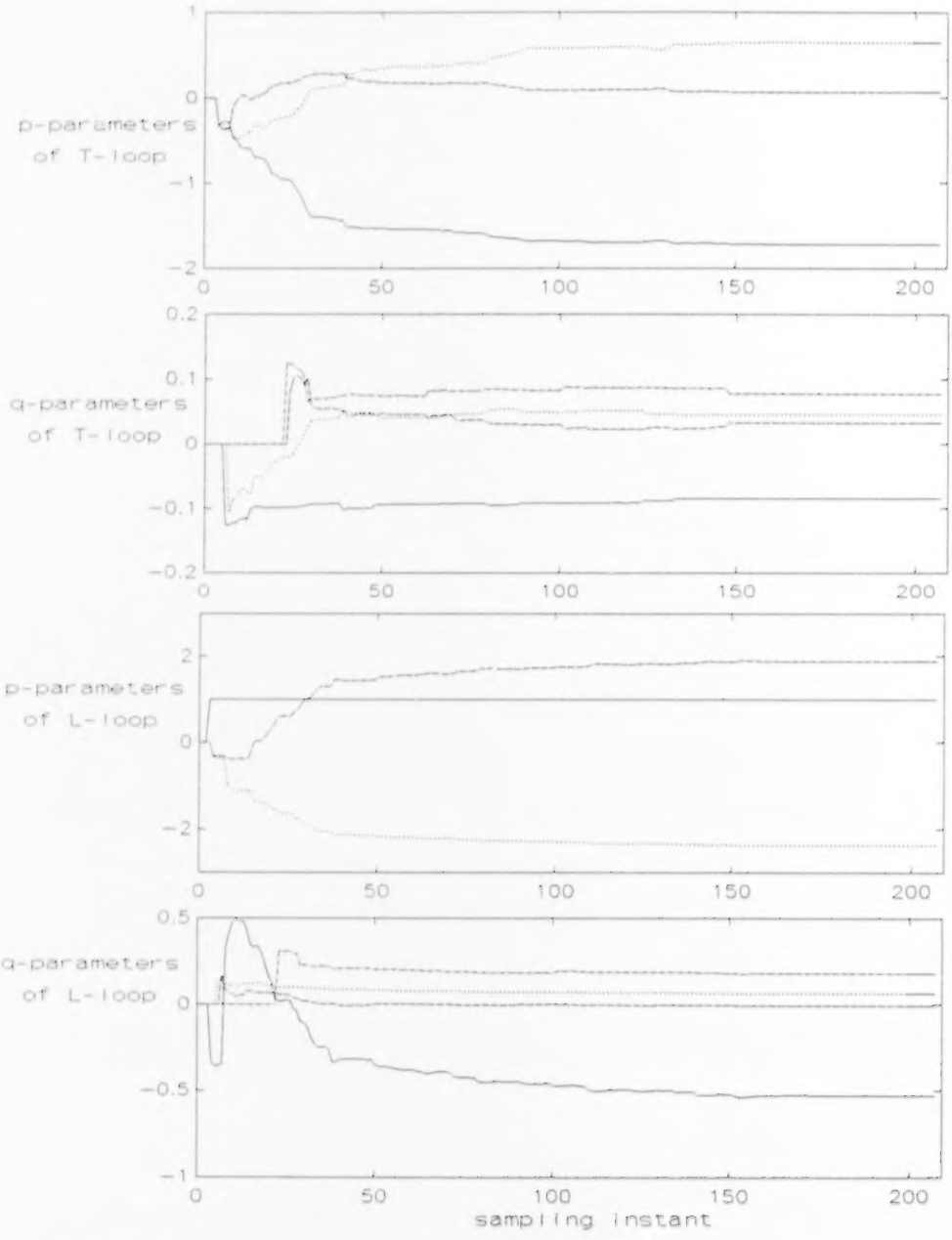


Figure 3.12 Parameter estimates of MISO subsystems corresponding to the temperature and level loops in ARIMAX representation without forgetting factor

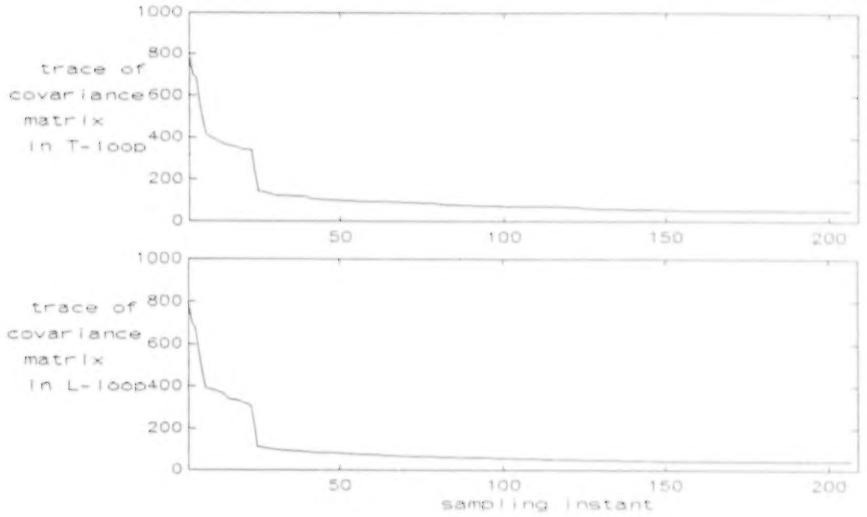


Figure 3.13 Trace of the covariance matrix of the data vector in both MISO subsystems in ARIMAX representation without forgetting factor.

The MIMO AUDI algorithm discussed in section 3.1.1 is now used for each MISO subsystem to identify the parameters in eqn 3.77. The values of the forgetting factor and the initial values of the covariance matrix are the same as in the ARMAX model identification. Here also the identified model parameters were validated by comparing the actual measured outputs with the estimated outputs from the ARIMAX model at every time step. The degree of accuracy or validation is shown in figure 3.11, including trajectories of the normalized errors between the actual and estimated outputs.

By comparing figure 3.11 with figures 3.5 and 3.6, it can be concluded, that the ARMAX model is better able to represent the process than the ARIMAX model. It appears that since only past estimates values of the output are used in the auto-regressive part, the errors in the ARIMAX model are integrated causing a large drift in the estimation errors.

3.2.1.3 MIMO AUD algorithm with constant forgetting factor (λ)

This is a variant of the basic MIMO AUD algorithm, by which relatively more weight is given to the more recent data compared to the old data. This is achieved by using a value of forgetting factor (λ) less than 1. The approximate window of past data used for identification is then equal to $1/(1-\lambda)$

According to this formula, if λ is nearer to 1, then the length of past data used is long as compared to λ nearer to 0.

Alertness of an identification algorithm is governed by the value of the covariance matrix. Figure 3.10 shows that the trace of the covariance matrices for both temperature and level loops rapidly decay to small values resulting in low gains for the parameter update equation or reduced alertness. By using a forgetting factor less than 1, this problem can be avoided. A forgetting factor value less than one causes the covariance matrix update to be inflated when the excitation is small or zero. This results in an alert algorithm, but can result in the covariance "windup". This is shown in figure 3.14 where the estimated outputs are now able to track the actual output even in the absence of excitation. The parameter estimates are shown in figure 3.15 and the estimates of the error of closure term is shown in figure 3.16. The "covariance windup" phenomena can be seen in figure 3.17. Large values of the covariance matrix may result in increased parameter variance resulting in estimated parameters that can be very sensitive to noise effects. This problem can be avoided by using a variable forgetting factor. This modification is discussed in the following section.

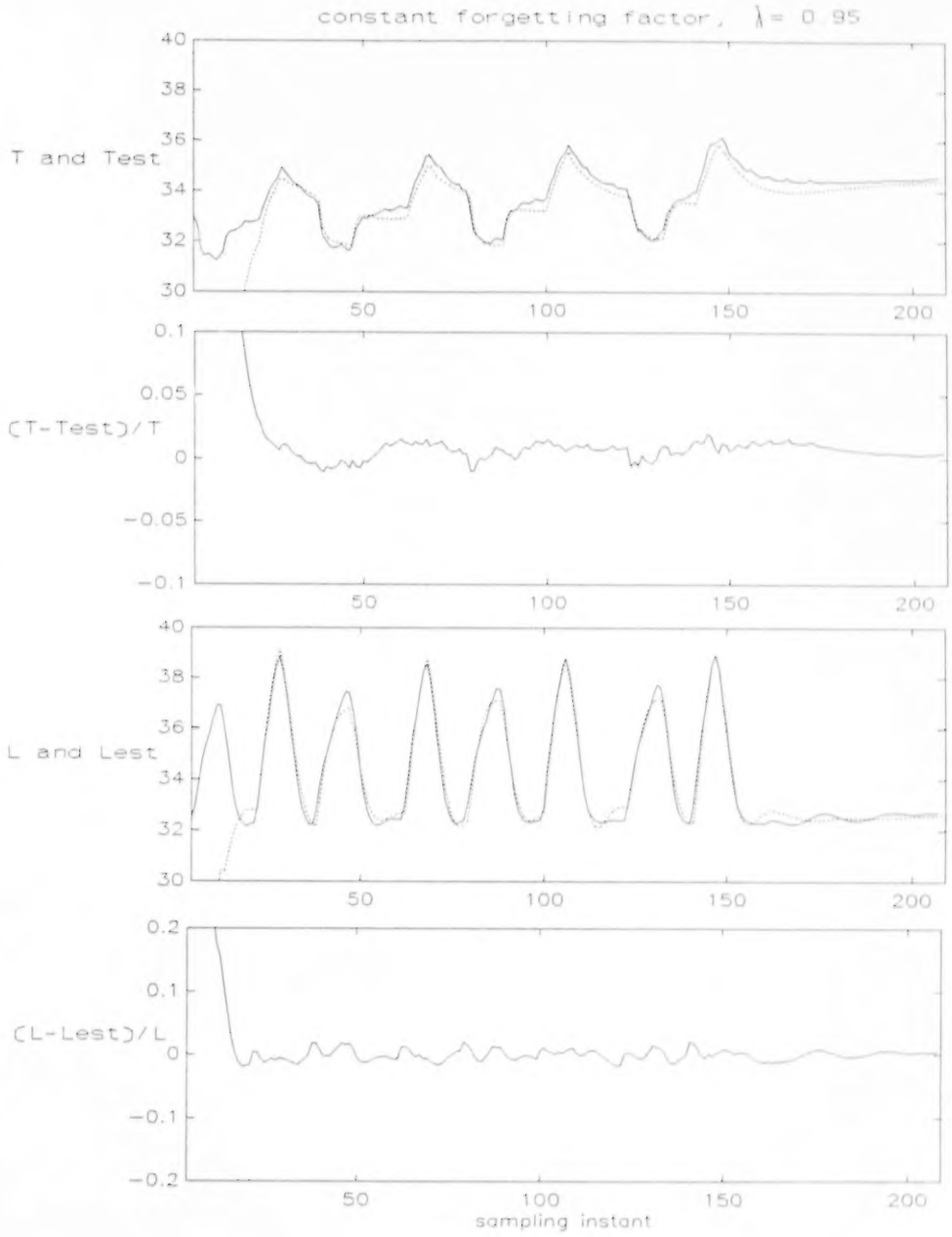


Figure 3.14 Actual and estimated temperature and level from ARMAX model identification with a constant forgetting factor of 0.95

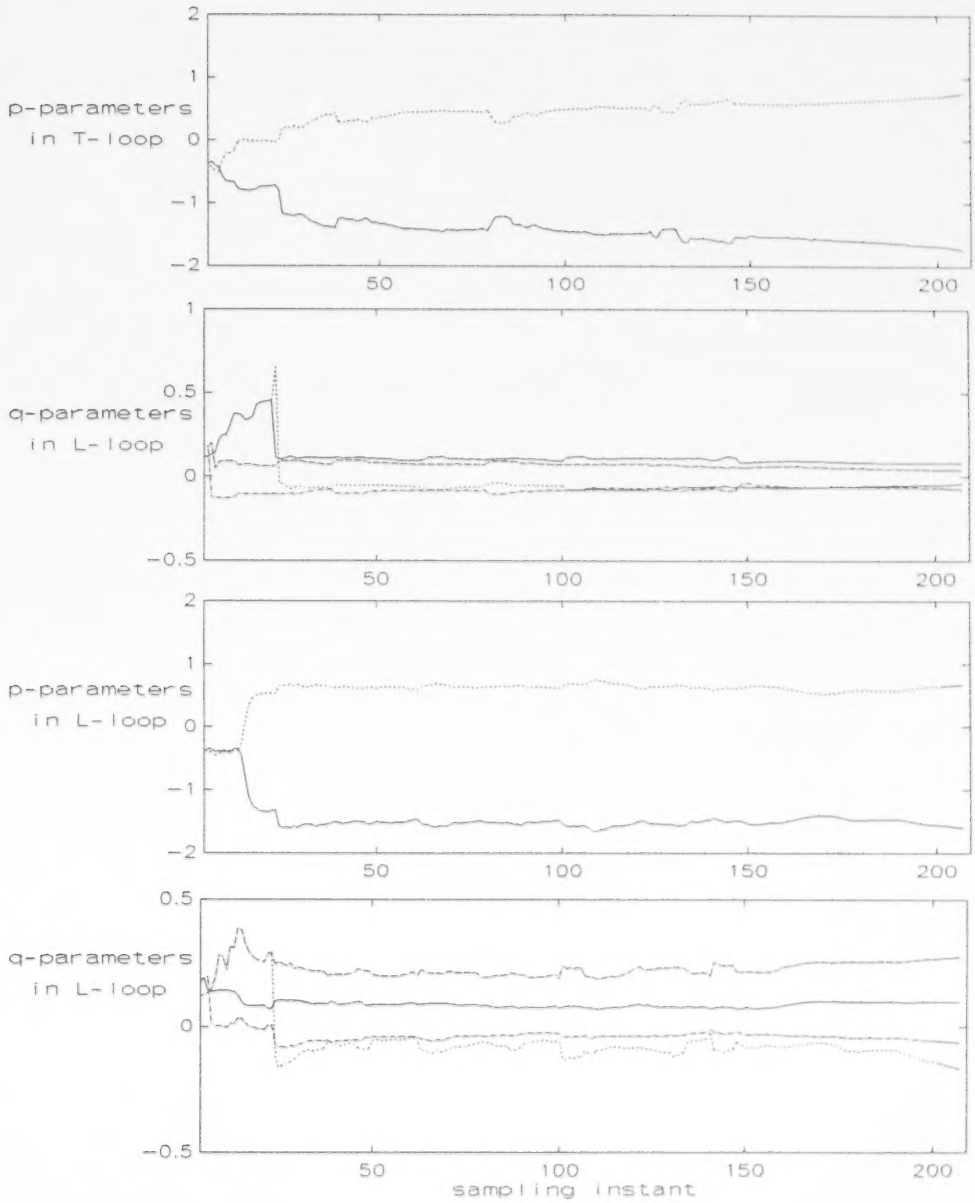


Figure 3.15 Parameter estimates of MISO subsystems corresponding to the temperature and level loops in ARMAX representation with a constant forgetting factor of 0.95

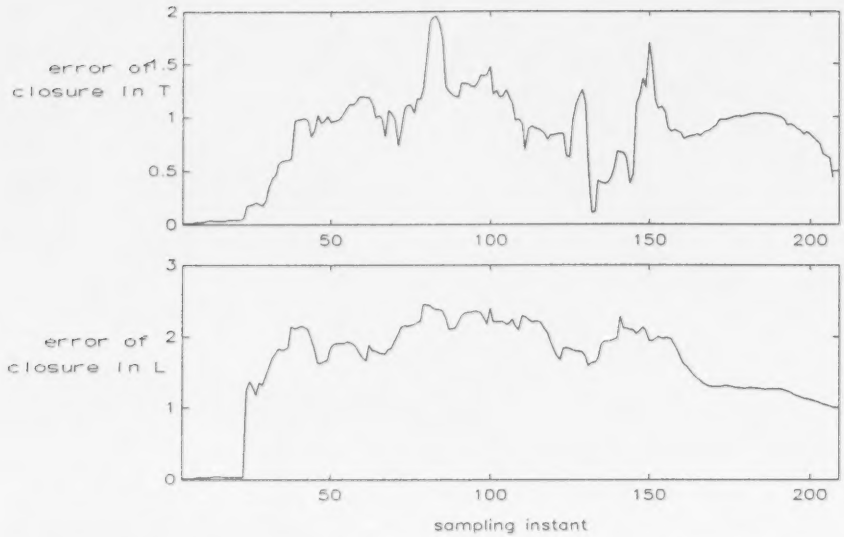


Figure 3.16 Estimates of the error of closure in temperature and level in ARMAX representation with a constant forgetting factor of 0.95.

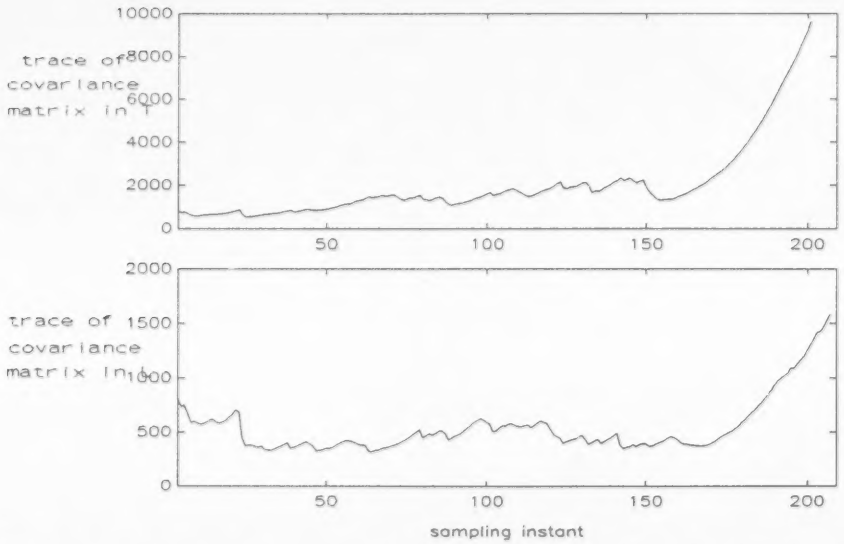


Figure 3.17 Trace of the covariance matrix of the data vector in both MISO subsystems in ARMAX representation with a constant forgetting factor of 0.95.

3.2.1.4 MIMO AUD algorithm with variable forgetting factor

In this variant of the algorithm, the value of the forgetting factor is calculated at every time step (Fortescue *et al.*, 1982). It is calculated as follows:

$$\lambda(t) = 1 - \frac{[y(t) - U(1:d-1,d)^T \phi(1:d-1)]^2}{\sigma[1 + f^T g]} \tag{3.78}$$

where,

$\sigma/\sigma_w \approx 100$, where σ_w is the variance of the measurement noise in the output. The value of σ is a very important tuning factor affecting the performance of the identification algorithm, since it determines the extent of forgetting that should occur.

In the present case, the values of σ_1 and σ_2 for both MISO subsystems were tuned by trial and error. For the values of $\sigma_1 = 0.012$ and $\sigma_2 = 0.02$, the MIMO AUD algorithm with variable forgetting factor was used to identify the ARMAX model of the process. Figure 3.18 shows the actual and the estimated outputs including the normalized estimation error. The estimated outputs are now able to track the actual output, even after the excitation is stopped. Figures 3.21 and 3.22 show the variation of the trace of the covariance matrix and the forgetting factor with respect to time. The values of the forgetting factor for both MISO subsystems at times dip down to values as low as 0.5, with the corresponding increase in the value of the covariance matrix trace. But now the values of the trace of the covariance matrix are not blowing up and thus the phenomena of "covariance windup" is avoided. This makes the algorithm alert but at the same time is able to keep the values of the trace of the covariance matrix from blowing up. Because of the alert identification, the algorithm will be sensitive to noise effects resulting in a

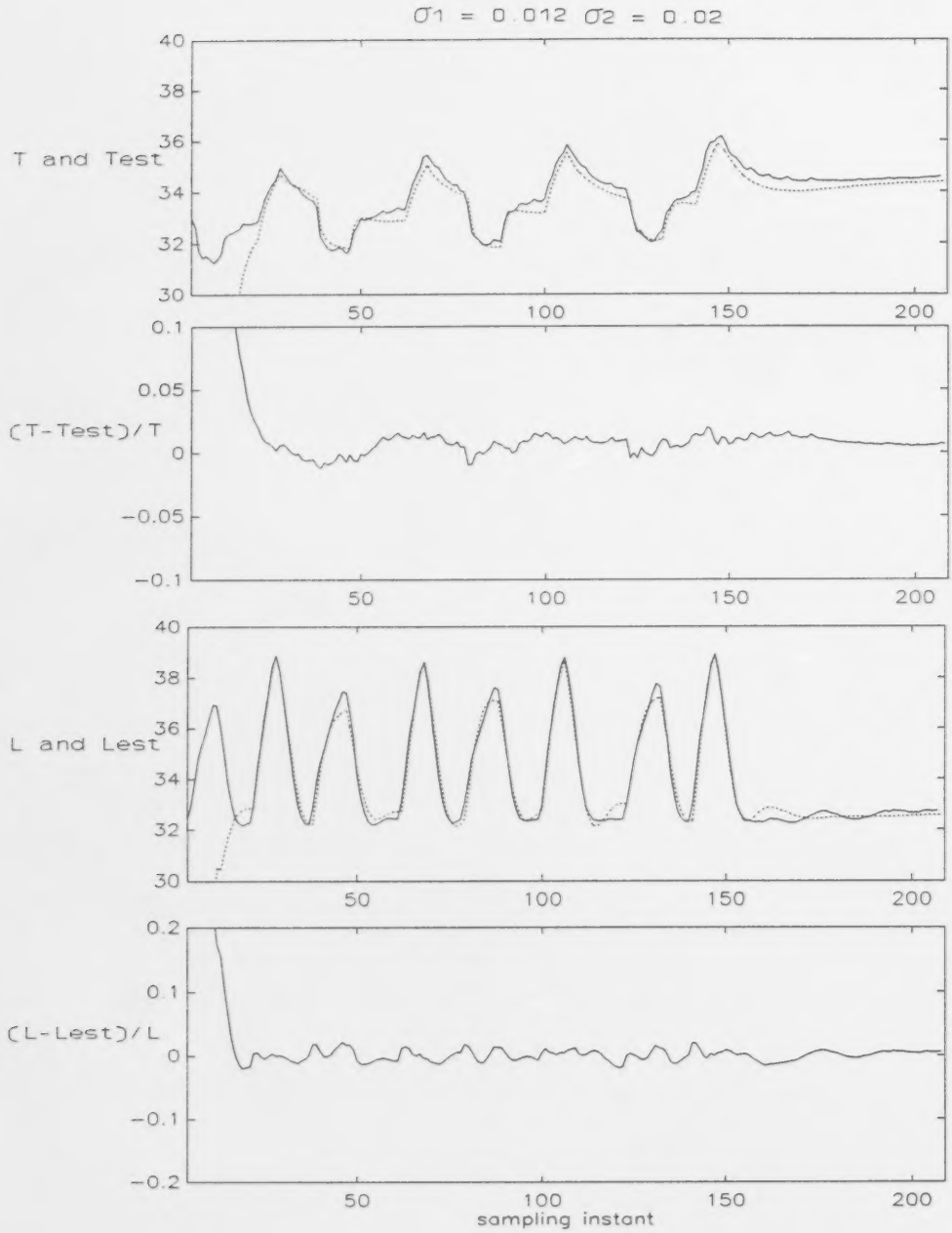


Figure 3.18 Actual and estimated temperature and level from ARMAX model identification with a variable forgetting factor.

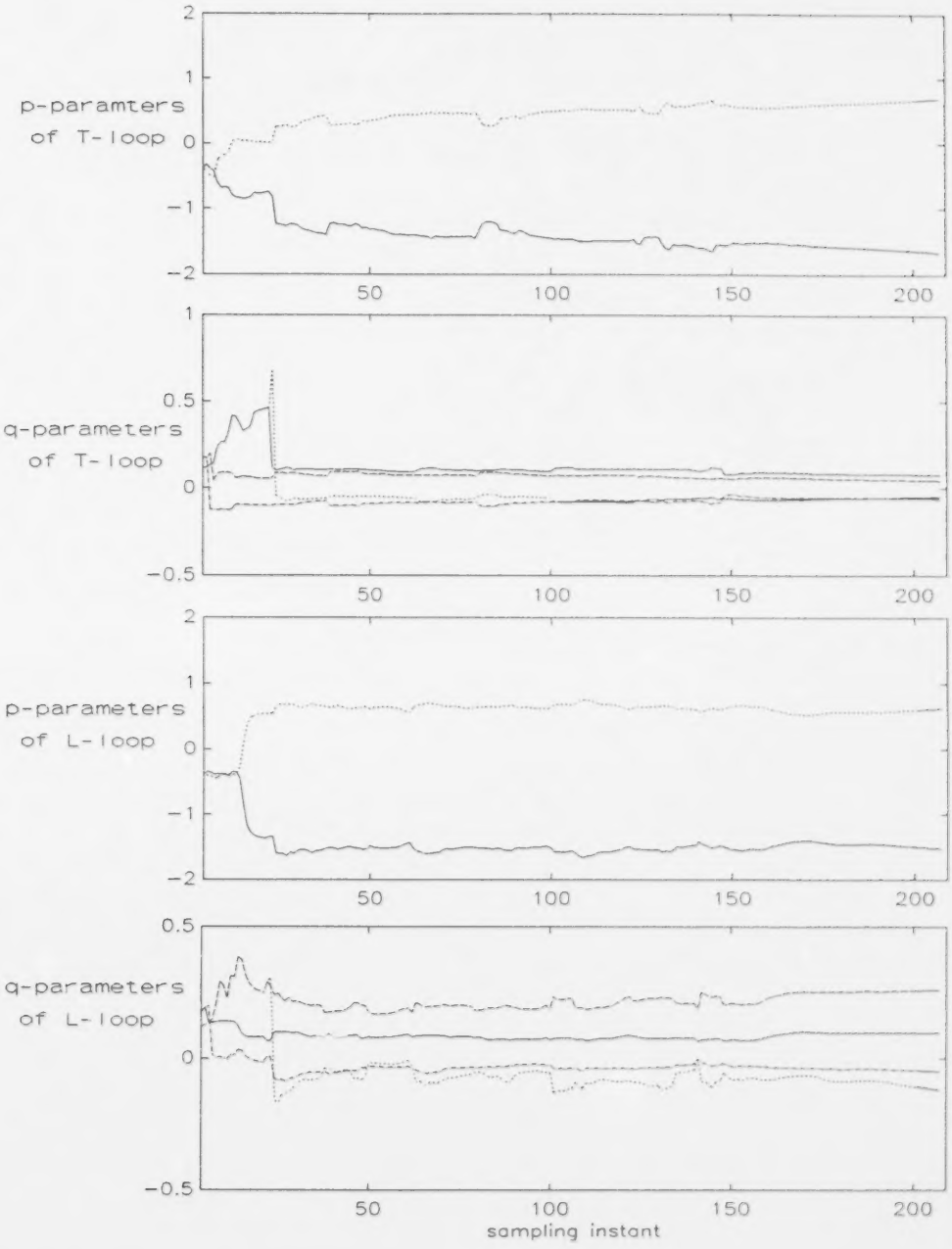


Figure 3.19 Parameter estimates of MISO subsystems corresponding to the temperature and level loops in ARMAX representation with a variable forgetting factor.

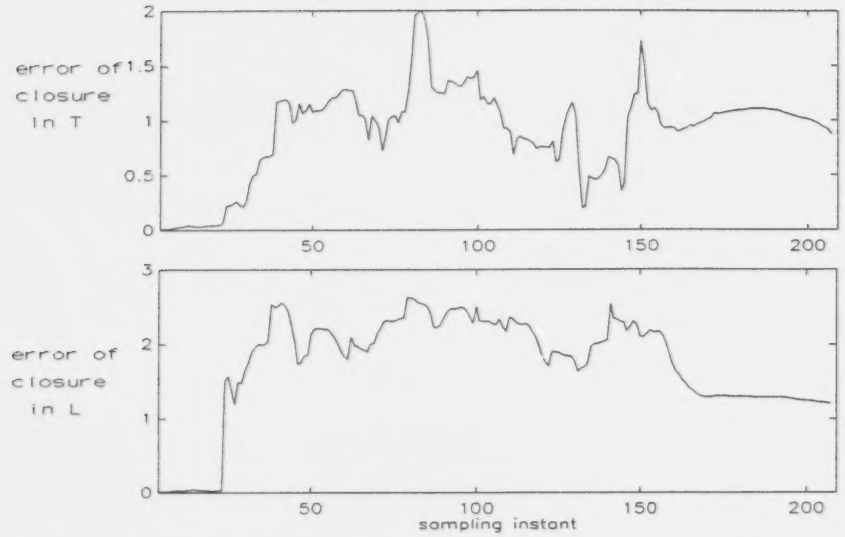


Figure 3.20 Estimates of the error of closure in temperature and level in ARMAX representation with a variable forgetting factor.

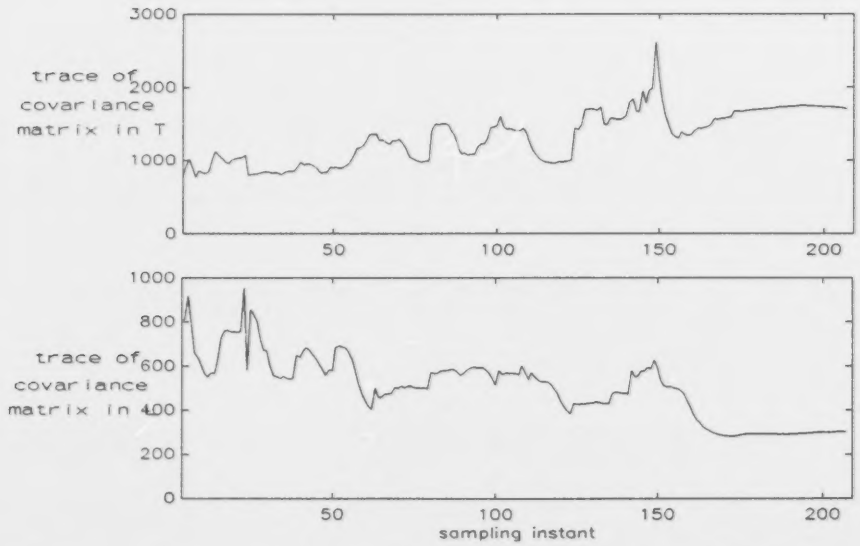


Figure 3.21 Trace of the covariance matrix of the data vector in both MISO subsystems in ARMAX representation with a variable forgetting factor.

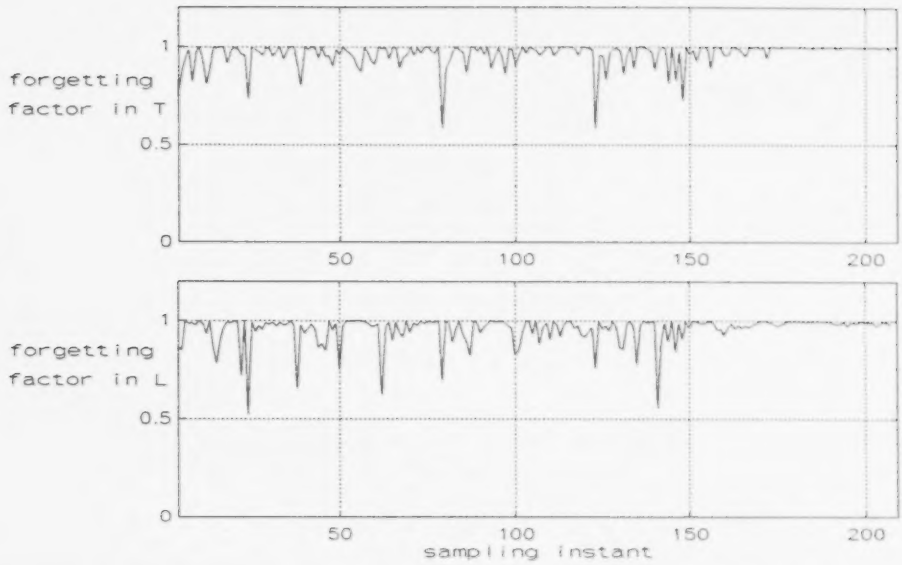


Figure 3.22 Variation of forgetting factor in ARMAX model identification

higher variance of the parameter estimates. Closed-loop adaptive control would therefore result in a large variance in the control action which is not advisable. To reduce the estimated parameter variance, it is advisable to bound the forgetting factor by a suitable lower bound.

Such bounds are incorporated into the MIMO AUD algorithm as follows:

if: $\lambda_1 <$ some lower bound λ_{lb}, then: $\lambda_1 = \lambda_{lb}$

The performance of the MIMO AUDI algorithm was evaluated with the lower bounds on the forgetting factor values set at $\lambda_{lb1} = \lambda_{lb2} = 0.75$. These results are presented in figures 3.23-3.27. It can be seen that the estimated outputs are able to track the actual outputs reasonably well, even in the absence of excitation. The parameter estimates now have smoother trajectories, i.e. the parameter estimates have a lower variance as compared to the unbounded variable forgetting factor shown in figures 3.19 and 3.20.

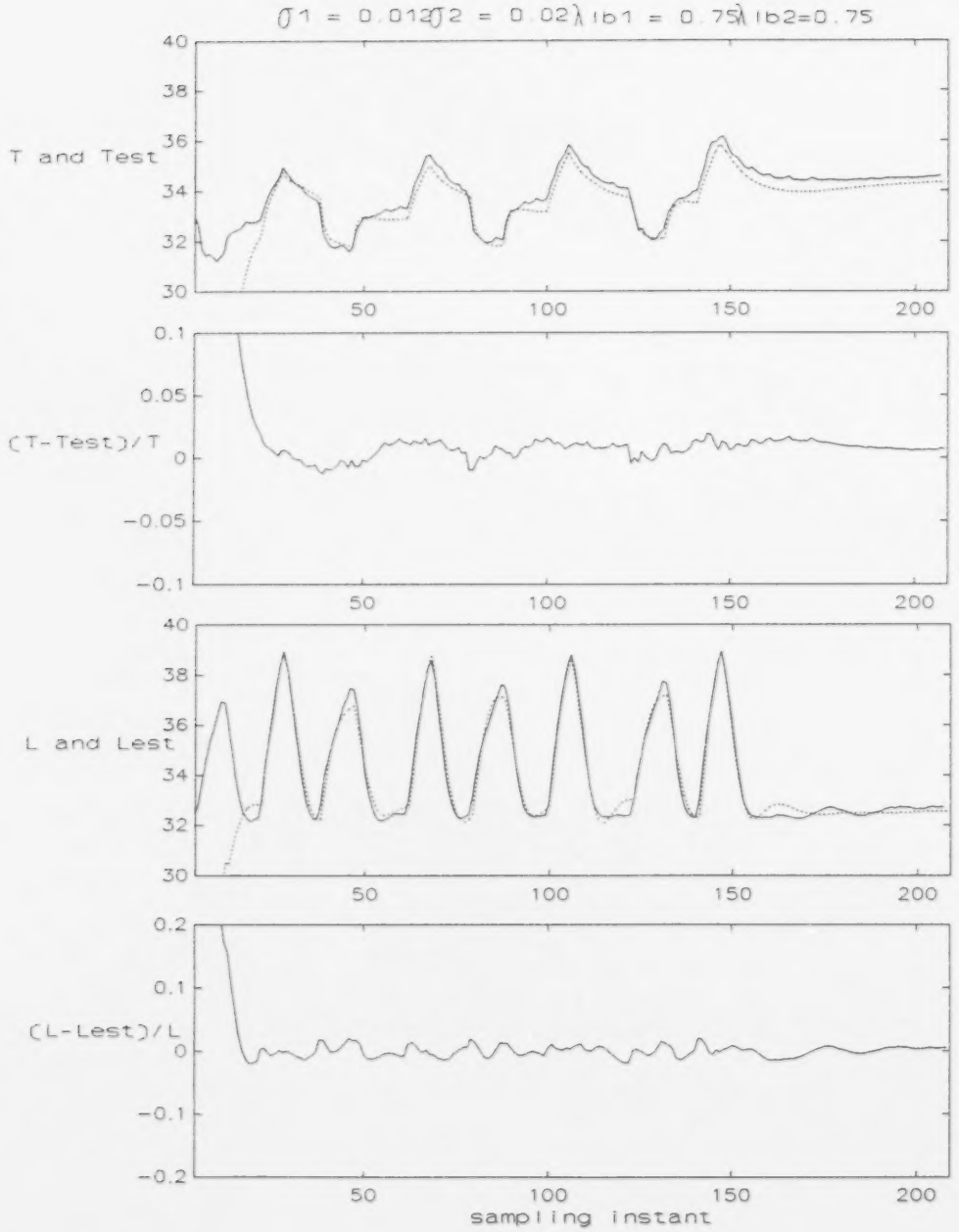


Figure 3.23 Actual and estimated temperature and level from ARMAX model identification using a variable forgetting factor with a lower bound at 0.75.

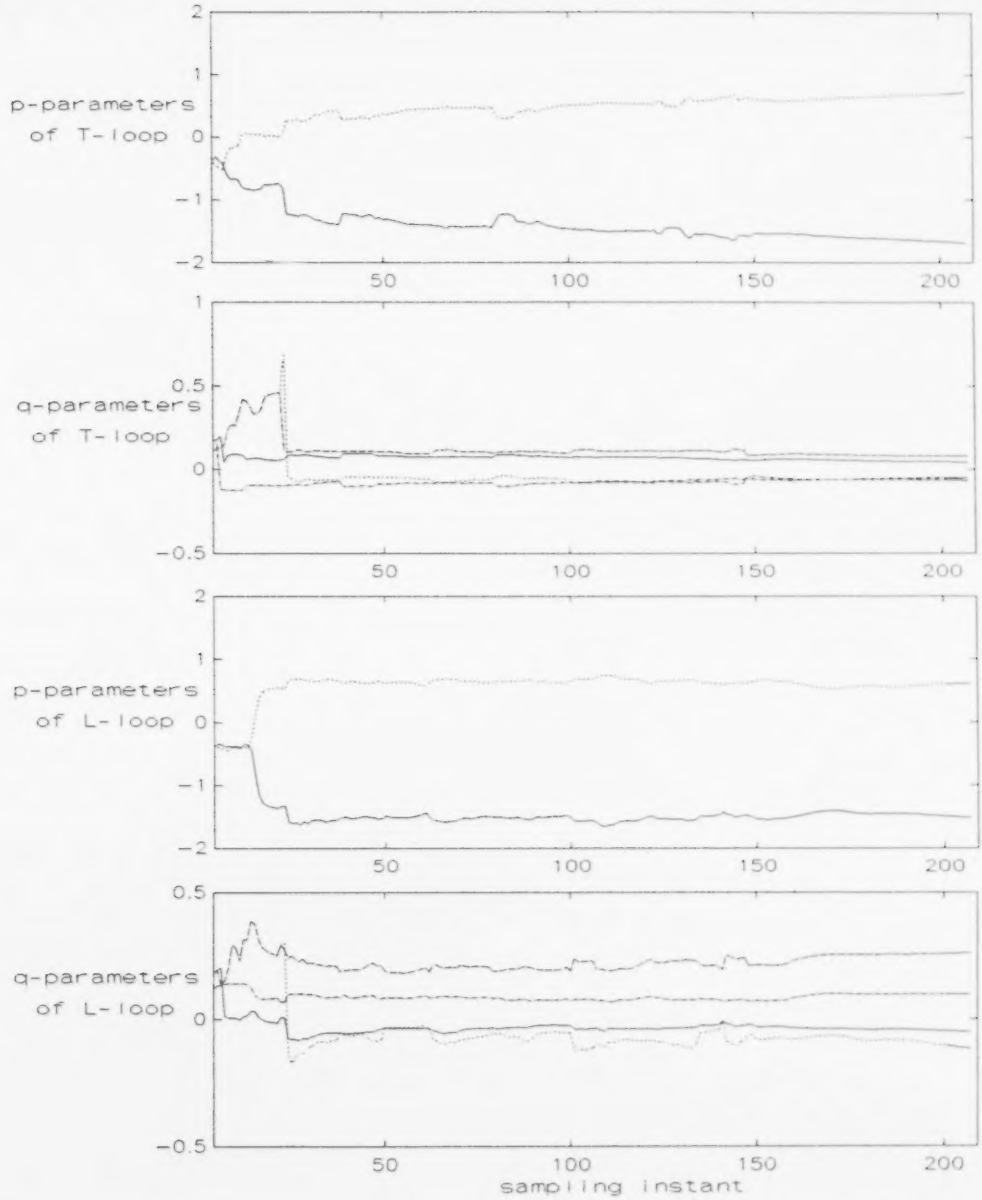


Figure 3.24 Parameter estimates of MISO subsystems corresponding to the temperature and level loops in ARMAX representation using a variable forgetting factor with a lower bound at 0.75.

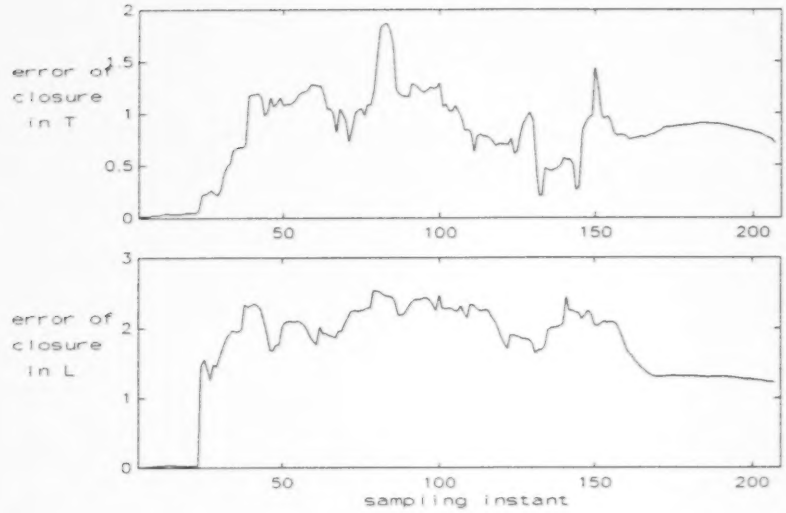


Figure 3.25 Estimates of the error of closure in temperature and level in ARMAX representation using a variable forgetting factor with a lower bound at 0.75.

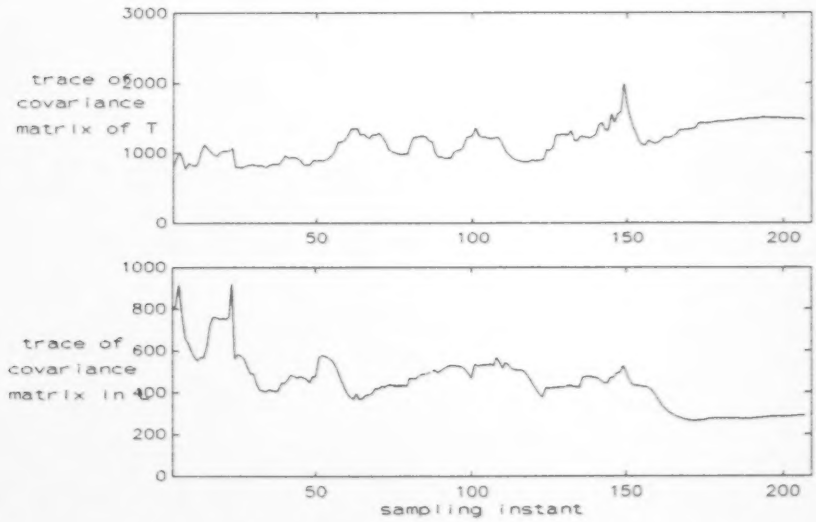


Figure 3.26 Trace of the covariance matrix of the data vector in both MISO subsystems in ARMAX representation using a variable forgetting factor with a lower bound at 0.75.

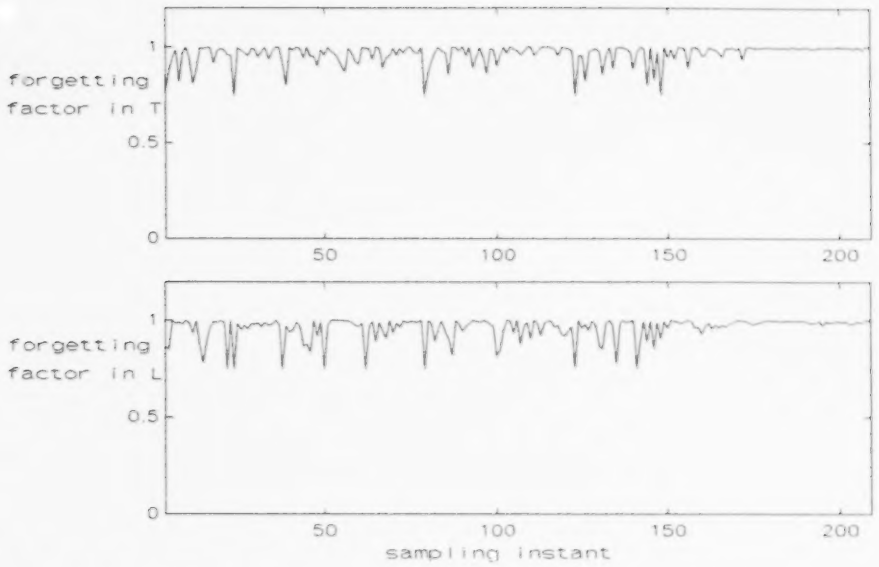


Figure 3.27 Variation of forgetting factor in ARMAX model identification.

In this chapter, a few popular techniques for recursive identification have been discussed. The AUD identification algorithm which has superior numerical properties has been extended to MIMO systems with distinct time-delays. The n x m multivariable system identification problem is first developed into an 'n' multi-input, single output (MISO) system identification problem.

Suitable modifications, such as variable forgetting factors with lower bounds, were proposed and successfully evaluated. These modifications enhance the alertness or adaptivity of the algorithm and yet at the same time give a reasonably smooth estimates parameter trajectories.

Chapter 4

Generalized Predictive Control of Multivariable systems

The Generalized Predictive Control (GPC) algorithm developed by Clarke *et al.*(1987), belongs to the class of generic long-range, multi-step predictive control algorithms. The multivariable extension of this algorithm (MGPC) has been described in detail by Mohtadi (1987) and Shah *et al.*(1987). The constrained version of MGPC with an analytical solution for a specific case of NU=1, was developed by Mutha (1990). This chapter briefly discusses the theory of the basic MGPC algorithm and its constrained version.

The basic GPC algorithm is capable of giving stable control of processes with variable parameters, variable time-delays, non-minimum phase characteristics and open-loop unstable poles (Clarke *et al.*, 1987).

4.1 The Multivariable GPC (MGPC)

In this section, the MGPC algorithm is developed based on the work of Mohtadi (1986), Shah *et al.* (1987) and Mutha (1990). In the development of the MGPC algorithm, an appropriate process model structure suitable for making future predictions of the process outputs has to be defined. This predictive model is then combined with a quadratic cost function, which when minimized gives the required control action.

4.1.1 The CARIMA or ARIMAX Process Model

The Controlled Auto-Regressive Integrated Moving Average model which is the same as the ARIMAX model (as discussed in sec 3.1.1), for an n-output m-input process can be defined as follows:

$$\underline{\underline{P}}(z^{-1})\ \underline{y}(t) \;=\; \underline{\underline{Q}}(z^{-1})\ \underline{u}(t-1) \;+\; \underline{\underline{\Delta}}^{-1}\underline{\underline{C}}(z^{-1})\underline{e}(t) \qquad (4.1)$$

where,

$\underline{\underline{P}}$ and $\underline{\underline{C}}$ are diagonal polynomial matrices of dimension n x n

$\underline{\underline{Q}}$ is a polynomial matrix of dimension n x m

$\underline{\Delta y}$ is a vector of differenced outputs of dimension n x 1

$\underline{\Delta u}$ is a vector of differenced inputs of dimension m x 1

$\underline{e}$ is a vector of random noise sequence of dimension n x 1

There can be distinct time-delays in different elements of the $\underline{\underline{Q}}$ polynomial matrix. If the time-delay is non-zero, then the leading elements in the polynomial will be zeros. $\underline{\underline{C}}$ is the polynomial matrix of the noise term. For the sake of simplicity it is taken to be *I*.

Thus the simple CARIMA representation of a process can be:

$$\underline{P}(z^{-1})\ \underline{y}(t) = \underline{Q}(z^{-1})\ \underline{u}(t-1) + \underline{\Delta}^{-1}\underline{e}(t) \tag{4.2}$$

This model representation is appropriate since this simple linear difference form can be readily manipulated into a predictive form. The assumed noise model can account for random steps occurring at random times or Brownian motion type disturbances. Such a noise model forces the use of incremental data in the control calculations due to the presence of an integrator in the model.

The updated parameters of the model are provided at every sample instant by the MIMO AUDI algorithm discussed in chapter 3.

4.1.2 Long Range Prediction in MGPC

In a long range predictive controller (LRPC), a series of future predicted outputs has to be computed based on the information from the process up to time t and assumptions about future control activity.

The long range prediction equation that predicts the outputs of a multivariable process can be formulated via the matrix Diophantine equation as shown below:

$$\underline{I} = \underline{\underline{E}}_j\underline{\underline{P}}\underline{\underline{\Delta}} + z^{-j}\underline{\underline{F}}_j \tag{4.3}$$

Multiplying equation 4.1 by $\underline{\underline{E}}_j\underline{\underline{\Delta}}z^j$, gives

$$\underline{\underline{E}}_j\underline{\underline{P}}\underline{\underline{\Delta}}\,\underline{y}(t+j) = \underline{\underline{E}}_j\underline{\underline{Q}}\underline{\underline{\Delta}}\,\underline{u}(t+j-1) + \underline{\underline{E}}_j\underline{e}(t+j) \tag{4.4}$$

Combining eqn 4.3 and eqn 4.4,

$$\underline{y}(t+j) = \underline{\underline{F}}_j \underline{y}(t) + \underline{\underline{E}}_j \underline{\underline{Q}} \Delta \underline{u}(t+j-1) + \underline{\underline{E}}_j \underline{e}(t+j) \quad (4.5)$$

$\underline{\underline{E}}_j(z^{-1})$ is a diagonal polynomial matrix and the degree of its polynomials is *j-1* (Clarke *et al.*, 1987). Thus, all the noise components in eqn 4.5 are in the future and the optimal output predictor based on the measured output data up to time *t* and any given *u(t+k) for* $k > 1$, will be:

$$\hat{\underline{y}}(t+j/t) = \underline{\underline{G}}_j(z^{-1}) \Delta \underline{u}(t+j-1) + \underline{\underline{F}}_j(z^{-1}) \underline{y}(t) \quad (4.6)$$

where, $\underline{\underline{G}}_j = \underline{\underline{E}}_j \underline{\underline{Q}}$

This can be rewritten as:

$$\begin{aligned} \hat{\underline{y}}(t+j) &= \tilde{\underline{\underline{G}}}_j(z^{-1}) \Delta \underline{u}(t+j-1) + \bar{\underline{\underline{G}}}_j(z^{-1}) \Delta \underline{u}(t-1) + \underline{\underline{F}}_j(z^{-1}) \underline{y}(t) \\ &= \tilde{\underline{\underline{G}}}_j(z^{-1}) \Delta \underline{u}(t+j-1) + \underline{f}(t+j) \end{aligned} \quad (4.7)$$

The prediction $\hat{y}(t+j)$ has been expressed as the sum of two parts:

$\underline{f}$, is the value of the outputs based on past values of the inputs, assuming all future changes in the inputs are zero. This is also called the *'free response'*; and

$\tilde{\underline{\underline{G}}}(z^{-1})\Delta\underline{u}(t+j-1)$ is the predicted effect of the future control moves on the outputs.

4.1.3 The Predictive Control Law of MGPC

The future predictions based on future control signals can be computed using eqn 4.6. The future control inputs, on the other hand are computed from the desired values of the future predictions. To derive the MGPC control law, a quadratic cost function which is the difference between future predictions of the outputs and a prespecified set-point

trajectory is defined, and the control law is derived by minimizing this cost function with respect to the changes in the control inputs.

The cost function minimized by the MGPC is as follows:

$$\begin{aligned} minJ_{MGPC} \;=\; & \sum_{j=N_1}^{N_2} [\hat{\underline{y}}(t+j) - \underline{w}(t+j)]^T [\hat{\underline{y}}(t+j) - \underline{w}(t+j)] \\ & + \sum_{j=1}^{N_2} \Delta \underline{u}(t+j-1)^T \Lambda \Delta \underline{u}(t+j-1) \end{aligned} \tag{4.8}$$

where,

N_1 is the minimum prediction horizon,

N_2 is the maximum prediction horizon,

$$\Lambda = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_m \end{bmatrix} \tag{4.9}$$

is a diagonal matrix with elements corresponding to different control weighting on each future incremental input.

The above cost function can be written in the following form, (please see Mohtadi (1986) and Shah *et al.*(1987)):

$$minJ_{MGPC} \;=\; [\underline{\underline{G}} U + \underline{f} - \underline{w}]^T [\underline{\underline{G}} U + \underline{f} - \underline{w}] \;+\; U^T \Delta U \tag{4.10}$$

On minimization, the solution is:

$$U \;=\; [\underline{\underline{G}}^T \underline{\underline{G}} + \Delta]^{-1} \underline{\underline{G}}^T (\underline{w} - \underline{f}) \tag{4.11}$$

where,

$\mathbf{U} = [\underline{\Delta u}(t)\ \underline{\Delta u}(t+1)\ \ldots\ \ldots\ \underline{\Delta u}(t + N_2 - 1)]^T$

$\underline{\mathbf{w}} = [w_1(t+1)\ \ldots\ w_n(t+1)\ \ldots\ \ldots\ w_1(t+j)\ \ldots\ w_n(t+j)]^T$

$\underline{\mathbf{f}} = [f_1(t+1)\ \ldots\ f_n(t+1)\ \ldots\ \ldots\ f_1(t+j)\ \ldots\ f_n(t+j)]^T$

$\underline{\mathbf{\Lambda}}$ = matrix of N_2 block diagonal Λ matrices

and the $\underline{\mathbf{G}}$ matrix of dimension $n(N_2 - N_1 + 1)\text{x}(m\ N_2)$:

$$\underline{\mathbf{G}} = \begin{bmatrix} g_{N_1-1} & \cdots & \cdots & g_0 & 0 & 0 & \cdots & \cdots & 0 \\ g_{N_1} & \cdots & \cdots & g_1 & g_0 & 0 & \cdots & \cdots & 0 \\ \vdots & \ddots & \ddots & & & & & & \vdots \\ g_{N_2-1} & \cdots & \cdots & & & & & & g_0 \end{bmatrix} \tag{4.12}$$

where each g is a submatrix of dimension n x m, the elements of which are the step responses of individual input-output transfer functions. Since GPC is a receding-horizon control strategy only the first control increment vector is implemented, i.e. $\underline{\Delta U}(t)$ is applied.

While implementing the MIMO adaptive GPC on a process, it is necessary to invert the matrix in eqn 4.11 at every time step. The dimension of this matrix is

$n\ N_2$ x $m\ N_2$ which will result in heavy computation at every sampling instant, since N_2 is usually a large number. This limitation can be overcome by specifying an extra tuning parameter called the control horizon (Mohtadi, 1986). This idea of specifying a control horizon, NU, has been borrowed from the dynamic matrix control (DMC) algorithm of Cutler and Ramaker (1980). It assumes that after NU future sampling instants, the projected control increments are taken to be zero.

The control horizon, NU, is the number of non-zero control increments the controller is

free to select for minimizing the cost function in eqn 4.8. By incorporating this idea the cost function to be minimized for MGPC is as follows:

$$minJ_{MGPC} = E\{ \sum_{j=N_1}^{N_2} [\hat{\underline{y}}(t+j) - \underline{w}(t+j)]^T [\hat{\underline{y}}(t+j) - \underline{w}(t+j)] + \sum_{j=1}^{NU} \Delta\underline{u}(t+j-1)^T \Lambda \Delta\underline{u}(t+j-1) \} \quad (4.13)$$

or

$$minJ_{MGPC} = [\underline{\underline{G'}} U + \underline{f} - \underline{w}]^T [\underline{\underline{G'}} U + \underline{f} - \underline{w}] + U^T \Delta U \quad (4.14)$$

On minimization, the solution is:

$$U = [I_m \, 0 \, \ldots\ldots] [\underline{\underline{G'^T G'}} + \Delta]^{-1} \underline{\underline{G'^T}} (w - f) \quad (4.15)$$

where,

$\mathbf{U} = [\underline{\Delta u}(t) \; \underline{\Delta u}(t+1) \; \ldots \; \ldots \; \underline{\Delta u}(t + NU - 1)]^T$

$\underline{w} = [w_1(t+1) \ldots w_n(t+1) \ldots \ldots w_1(t+j) \ldots w_n(t+j)]^T$

$\underline{f} = [f_1(t+1) \ldots f_n(t+1) \ldots \ldots f_1(t+j) \ldots f_n(t+j)]^T$

$\underline{\Lambda}$ = matrix of NU block diagonal Λ matrices

and the $\underline{\underline{G}}$ matrix of dimension $n(N_2 - N_1 + 1)x(m\ NU)$:

$$\underline{\underline{G'}} = \begin{bmatrix} g_{N_1-1} & \cdots & \cdots & g_0 & 0 & 0 & \cdots & \cdots & 0 \\ g_{N_1} & \cdots & \cdots & g_1 & g_0 & 0 & \cdots & \cdots & 0 \\ \vdots & \ddots & \ddots & & & & & & \vdots \\ g_{N_2-1} & \cdots & \cdots & & & & & & g_{N_2-NU} \end{bmatrix} \quad (4.16)$$

4.1.4 Constraints in MGPC algorithm

The control action implemented on a real process can have limitations due to saturation of actuators, i.e. a valve can only be opened between the range of 0 % and 100 %. There can also be physical limitations or quality constraints on the outputs of the process, i.e. temperature measurement can only be possible within a particular range or some quality standards have to be met in terms of concentration of the product from a process. These physical limitations need to be incorporated in the control calculations so that only physically realizable control actions will be calculated by the control law.

The constraints of the real processes are on the amplitude of the control action implemented, i.e. a limitation on the absolute value of the control input that can be implemented and on the incremental control values and on the values of the process outputs.

For NU=1, the amplitude constraints on the process inputs can be expressed as constraints on the $\underline{\Delta u}$ as follows: (Mutha, 1990).

$$\begin{aligned} &if \quad \underline{u}_{min} \le \underline{u}(t) \le \underline{u}_{max} \\ &\therefore \quad \underline{u}_{min} - \underline{u}(t-1) \le \underline{\Delta u} \le \underline{u}_{max} - \underline{u}(t-1) \end{aligned} \tag{4.17}$$

For any general NU:

$$\begin{bmatrix} \underline{u}_{min} - \underline{u}(t-1) \\ \underline{u}_{min} - \underline{u}(t-1) \\ \vdots \\ \underline{u}_{min} - \underline{u}(t-1) \end{bmatrix} \le \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 1 & 1 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix} \begin{bmatrix} \underline{\Delta u}(t) \\ \underline{\Delta u}(t+1) \\ \vdots \\ \underline{\Delta u}(t+NU-1) \end{bmatrix} \le \begin{bmatrix} \underline{u}_{max} - \underline{u}(t-1) \\ \underline{u}_{max} - \underline{u}(t-1) \\ \vdots \\ \underline{u}_{max} - \underline{u}(t-1) \end{bmatrix} \tag{4.18}$$

The constraints on the process outputs can also be mapped into constraints on $\underline{\Delta u}$ as:

$$\underline{y}_{min} \leq \hat{\underline{y}} \leq \underline{y}_{max}$$
$$\therefore \underline{y}_{min} - \underline{f} \leq \underline{\underline{G}}\,\underline{\Delta u} \leq \underline{y}_{max} - \underline{f} \tag{4.19}$$

It should be noted here, that the mapping of the output constraints involves a pseudo-inverse of the step-response matrix $\underline{\underline{G}}$ (eqn 4.19). Thus translation of output constraints as input constraints is not exact, it is good only in a least squares sense.

The solution of the constrained MGPC can be obtained by minimizing the cost function (eqn 4.13 or eqn 4.14) subject to the inequality constraints described in eqn 4.18 and eqn 4.19. Commercial optimization software packages such as QPSOL (Gill *et al.*, 1984) and TOLMIN (Powell M.J.D, 1989) are available, and can be used to solve the constrained problem.

For the general case of MGPC algorithm, an analytical solution of the constrained cost function does not exist and has to be calculated by using any one of the commercial optimization software packages. For some special cases, the constrained cost function of GPC has been solved analytically, eg.

1. Tsang and Clarke (1988) have analytically solved rate or amplitude constrained SISO GPC for special cases of NU=1 and NU=2.
2. Mutha (1990) has analytically solved rate, amplitude and output constrained MGPC for the special case of NU=1. An analytical solution for the rate and amplitude constraints were also solved for the SISO GPC for NU=1 and 2.

The effects these constraints on the performance of MIMO Adaptive GPC are discussed in chapter 5.

4.1.5 Output Weighting in MGPC

The output weighting parameter in the cost function is relevant only in multivariable systems. This parameter plays an important role in the performance of the MAdGPC algorithm.

The MGPC cost function with the output weighting term can be expressed as follows:

$$\begin{aligned} min J_{MGPC} \;=\; & E\{ \sum_{j=N_1}^{N_2} [\hat{y}(t+j) - \underline{w}(t+j)]^T Y_{WT} [\hat{y}(t+j) - \underline{w}(t+j)] \\ & + \sum_{j=1}^{N_2} \Delta \underline{u}(t+j-1)^T \Lambda \Delta \underline{u}(t+j-1) \} \end{aligned} \tag{4.20}$$

where,

$$Y_{WT} = \begin{bmatrix} y_wt_1 & 0 & \cdots & 0 \\ 0 & y_wt_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & y_wt_n \end{bmatrix} \tag{4.21}$$

On rearrangement, eqn 4.21 can be written as follows:

$$\begin{aligned} \min J_{MGPC} \;=\; & \sum_{i=1}^{n} \left(\sum_{j=N_1}^{N_2} y_wt_i [y_i(t+j) - w_i(t+j)]^2 \right) \\ & + \sum_{j=1}^{N} \Delta u(t+j-1)^T \Lambda \Delta u(t+j-1) \end{aligned} \tag{4.22}$$

This parameter provides an option to declare which output variables should have a higher priority in terms of regulation. By this a 'symmetric' control performance can be ensured

by scaling the individual loops of MGPC such that under typical operating conditions, the values used within the control calculation after analog to digital conversion and normalization are of similar magnitude. Decoupling of a particular loop can be enhanced by weighting the corresponding output significantly more than the others, thus decoupling that output from the set point changes in the other loops (Maurath *et al.*,1986).

Chapter 5

Experimental Evaluation of Multivariable Adaptive GPC

Although evaluation of advanced control algorithms by simulation provides insight into the performance of the algorithms under controlled conditions, there is no substitute for evaluation of the algorithms under real conditions. By applying the algorithms on a real process, the performance of the algorithm can be tested in the presence of unmeasurable disturbance effects such as noise, load disturbances and unmodelled dynamics as well as nonlinearities.

The strategy for identification of a multivariable system using the MIMO augmented UD identification (MIMO AUDI) algorithm was discussed and developed in chapter 3. Its performance was studied by using input-output data from a real process. The non-adaptive MIMO GPC algorithm was discussed in chapter 4. In this chapter these two

algorithms are used together to implement and evaluate a MIMO adaptive GPC (MAdGPC) on a pilot-scale continuous stirred tank heater.

This chapter is organized as follows:

Section 5.1 provides the implementation details of the hardware used, the QNX operating system on which the algorithms are executed, details of the MULTICON software package which is used to administer various tasks, and for data-acquisition and a description of the process. In the section 5.2 the experimental runs obtained while implementing MAdGPC algorithm are presented.

5.1 Implementation Details

5.1.1 Hardware details

The multivariable identification algorithm and the multivariable GPC algorithm were implemented on an IBM PS/2 model 70 personal computer running under the QNX operating system. The computer communicates with the process through the Opto-22 interface. The Opto-22 behaves as a low end computer to the host computer (IBM PS/2 model 70) connected through a RS-232 serial line.

Its responsibility includes:

1. optically isolating the process from the computer,
2. continuously scanning the I/O channels and providing signal conversions for inputs and outputs (eg. analog to digital conversion), and
3. responding to requests from the host computer.

Whenever the host computer needs to access a process variable, it sends a command to

Opto-22. The Opto-22 sends the most recent digital data from the appropriate transducer and loads it onto the serial line to send the data to the host computer. When the host computer has to send a signal to the process, for example, the most recently calculated control action is sent via a serial line to the Opto-22, where the conversion of the signal from digital to analog takes place and then the signal is sent to the final control element. A complete set of communication protocols to operate Opto-22 is provided in the Optomux Analog User's Manual.

5.1.2 The QNX Operating System

QNX is a PC based real-time, multiuser, multitasking operating system marketed by Quantum Software System Limited of Ottawa. QNX uses a modular architecture that is based on message passing. It does not require much memory and has turned out to be very flexible in terms of real-time control applications.

5.1.2.1 Structure of QNX in brief

QNX is built on a client-server model. The approach is to implement most of the OS functions in server tasks and link them up through a message system. The operating system can be easily extended by connecting the user-defined tasks through messages. The MULTICON software package is an example of such a task. At the startup, 5 server tasks are created. Each of these server tasks provides a set of system calls for the user to use the hardware of the computer.

These tasks are:

1. Task administrator (task):

It is responsible for creating and destroying tasks and for allocating memory for the tasks. This is the most important task in the whole system.

2. Device administrator (dev):

It is responsible for handling all requests to open, close, read and write to character devices such as the terminal, keyboard, line printer etc.

3. File system administrator (fsys):

It is responsible for handling all requests to open, close, read and write to files.

4. Network administrator (net):

It is responsible for handling all data requests which must be transmitted over the network. This task exists only in the network version of QNX.

5. Idle administrator (idle):

It is responsible for consuming any spare processor time.

There are 2 other important tasks which are not created when the machine is booted, but can be created later. These are:

1. The Timer :

This can be created after the system is running. It gives timing capabilities to the system and is responsible for providing "wake up" service for real-time applications.

2. The Shell :

This task is created at log in. Although it is not a part of the QNX operating system, just like the test editor and the compiler, it serves as a primary interface between the user and QNX. It uses a terminal as the standard output and a keyboard as the standard input.

When a user types in a command to start a user program, the shell creates a task and runs the user program, allocating all the resources required to run the task. Once the task is completed, it reclaims all the resources that were allocated to the task.

For a single processor computer to accomplish multi-tasking, the operating system must be able to switch the CPU and the other resources for the use of several time-competing tasks. This scheduling is done by the kernel in QNX. It uses the "round robin" algorithm to schedule the tasks.

The rules are as follows:

1. When a task is waiting for input or output, it will not occupy any CPU time.
2. Each task can occupy the CPU only for a certain period of time (50 millisec) if there is another task waiting to run. All tasks are queued in a round robin fashion.

QNX also supports priority scheduling. Each task is assigned a priority, and the runnable task with the highest priority is allowed to run.

Further details on QNX and its features can be obtained from the manuals provided by the Quantum Software service Ltd. and from documentation by Lau (1990).

5.1.3 Multipurpose Control System (MULTICON) package

Multicon is a real time executive program that runs under the QNX operating system. It was developed by Qiu (1988) at the Department of Chemical Engineering, University of Alberta. It was created to support the following:

1. Execution of user tasks and to supervise their activity for real-time control applications.

2. Creation of a channel for process operator communication.

3. Provide a common data area for all the user tasks and a set of utility programs to access the data.

4. Display on-line graphics tasks.

5. Perform task scheduling.

6. Handle I/O device drivers.

Since, MULTICON is built on the QNX operating system, true multi-tasking is possible, i.e. more than one task can be run concurrently (eg. a task for multivariable identification can concurrently run with a task for multivariable GPC, which is being done in the experiments described in the subsequent sections). The user has only to create and define all the tasks in MULTICON and they will be scheduled properly. MULTICON greatly simplifies the communication between different tasks running under it. With all these features and capabilities, MULTICON is a very flexible shell for a control engineer to develop, test, and compare different control algorithms.

5.1.3.1 Structure of MULTICON in brief

The MULTICON package is made up of five independent tasks which communicate with each other by message passing:

1. Multicon (core) :

This is the main administrative task and is responsible for creating the other four system tasks as well as the user defined tasks at the start of the session and for terminating them when the session ends.

2. Timing :

This task is responsible for giving soft interrupts (or clock ticks) to the core *multicon* task at a resolution of one second. This is used by the core to check if there is any task that needs a "wake up" call.

3. Opto_drv :

This task is responsible for establishing a link between the *multicon core* and the "outside world" through the serial port. It is created by the *multicon core* but is run in the background. At startup, this task initializes the opto-22 interface and configures the I/O channels. Then it asks *multicon* regularly if there is any process variable to be accessed from the process.

4. Sys_poc :

This is the user interface of MULTICON. Its main functions are to do maintenance of the database and to interface the user with the *multicon core*. It is created by the *multicon core* and it runs concurrently with it. It scans the keyboard in a continuous fashion for a key character and performs accordingly.

5. Graph :

This task gives MULTICON its on-line graphics feature. It is created by the *multicon core* to run on another window of the same node or on another node on network, so that the user can switch easily between the graphics window and the main control panel. For more detailed information on MULTICON please refer to Lau (1991) and Qiu (1988). All the experimental results described in this chapter were conducted with the above described QNX operating system, MULTICON and the process interface Opto-22.

5.2 Adaptive Control of Continuous Stirred Tank Heater using MIMO Adaptive GPC algorithm

To evaluate the performance of MIMO Adaptive GPC (MAdGPC) algorithm, experiments were conducted on a continuous stirred tank heater (CSTH). The subsection 5.2.1 describes the process and its configuration for the experiments. The subsection 5.2.2 gives a brief description of how the MIMO AUD algorithm and the MIMO GPC algorithms are run simultaneously to implement the adaptive controller. In subsection 5.2.3, the results obtained from MAdGPC algorithm by experimental implementation on the continuous stirred tank heater are presented. This is followed by a discussion of the effects of various tuning parameters and guidelines for using them to adjust the performance of the controller.

5.2.1 Process Equipment

The MIMO configuration of the continuous stirred tank heater is shown schematically in figure 5.1. The equipment has two water lines carrying cold and hot water respectively entering the glass tank. The supply of the hot and the cold water is controlled by the Utility services of the University of Alberta. The temperature of the cold water varied between 10 to 15 degrees centigrade and the temperature of the hot water varies between 45 to 50 degrees centigrade. The water from the glass tank exits through a long copper tube, which has thermocouples at varying distances.

The nominal operating conditions are as follows:

Inlet cold water temperature	10 ° centigrade
Inlet hot water temperature	50 ° centigrade
Cold water valve opening	15 %
Hot water valve opening	10 %
Exit water temperature	33 ° centigrade
Level of water in tank	30 % of tank height

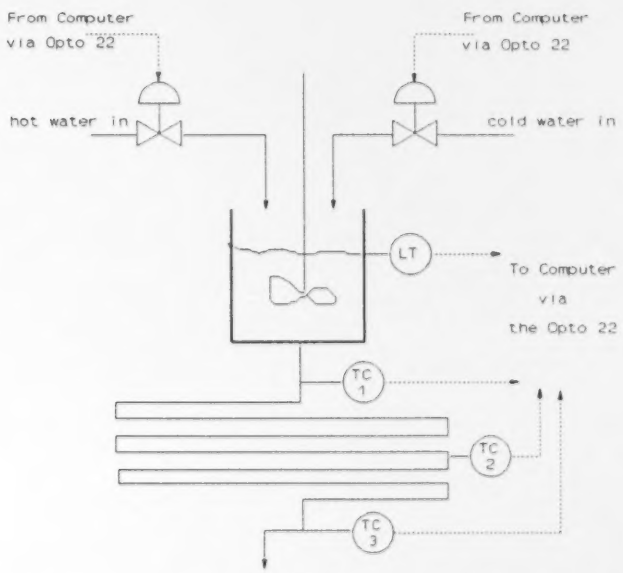


Figure 5.1 Schematic representation of the Continuous stirred tank heater for MIMO configuration

5.2.2 Implementation of MIMO Adaptive GPC

The two main elements of an adaptive controller, are the identification block and the control block. In this case the MIMO AUDI algorithm and the MIMO GPC algorithm for control when implemented together forms the MIMO adaptive GPC (MAdGPC) algorithm. It is important to discuss how these two algorithms can be combined to implement the MAdGPC.

The MIMO GPC is a model-based long-range predictive control algorithm. When this control algorithm is implemented using a constant model of the process, then MIMO GPC is said to run in a non-adaptive mode. But if there is an algorithm which can provide on-line parameter estimates for the process model at every time step to the MIMO GPC control algorithm, then it will run in an adaptive mode. Figure 5.2 shows the schematic representation of such an adaptive controller.

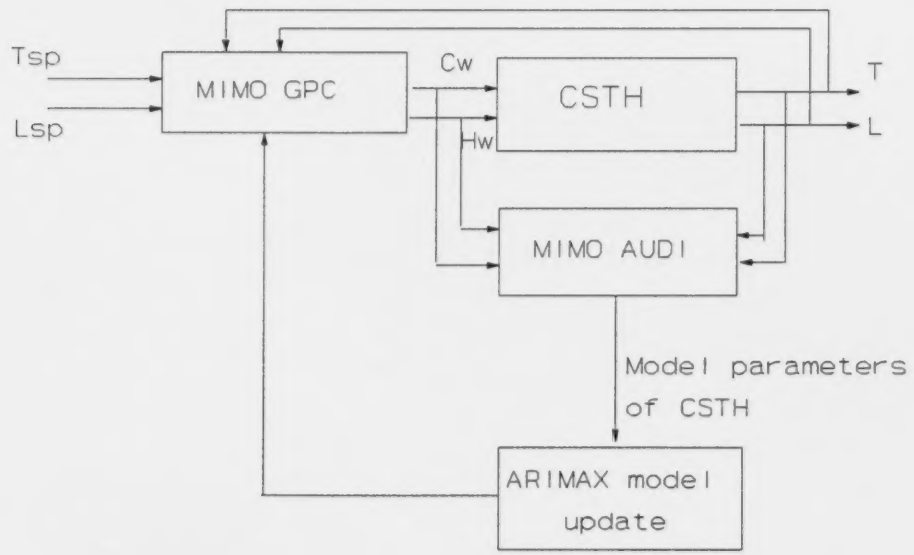


Figure 5.2 Schematic representation of MIMO Adaptive GPC

The MIMO AUD algorithm was coded in C programming language and the software for MIMO GPC algorithm was developed by Mutha (1990). These two algorithms run as separate tasks within MULTICON, operating under the QNX operating system. MULTICON stores the values of the measured variables and the previous inputs and makes them available to various tasks that need these values. Both these algorithms communicate between each other through message passing via MULTICON. The model parameters identified by the MIMO AUD algorithm are made available to the MIMO GPC algorithm for controller calculations. The results of the control calculations are passed to the OPTOMUX via the appropriate tasks within MULTICON. Figure 5.3 shows this in a schematic form.

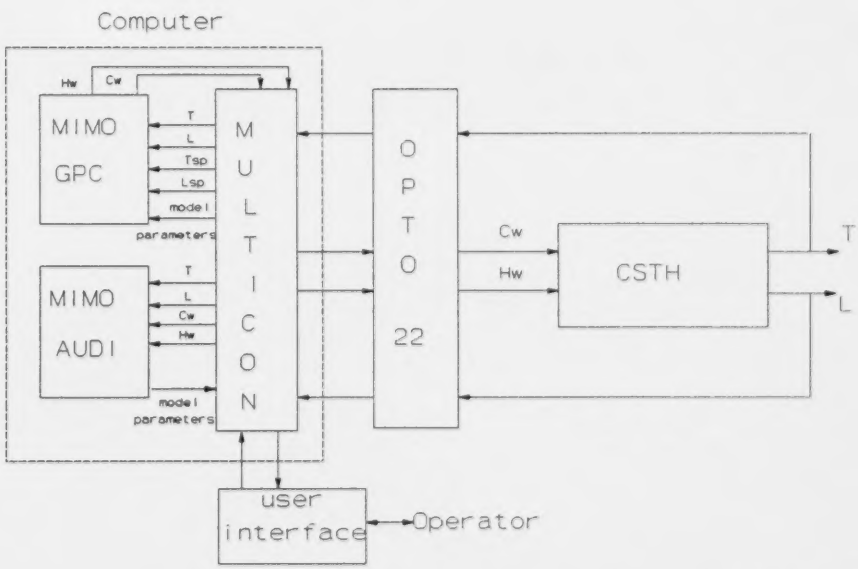


Figure 5.3 Schematic representation of the implementation of the MIMO adaptive GPC on continuous stirred tank heater.

5.2.3 Experimental Runs on Continuous Stirred Tank Heater

The MIMO adaptive GPC was implemented on the CSTH with the temperature sensor located at thermocouple # 1. The model parameters of the process were identified using the MIMO AUD identification algorithm. The ARIMAX model required by the MIMO GPC algorithm was updated at every time instant, with a sampling time of 4 seconds, using the latest values of the model parameters. This section demonstrates the effects of various tuning parameters of the adaptive controller. The performance of the MAdGPC was evaluated under a variety of different conditions and the results are presented in this section. The performance of the MAdGPC was also compared with the performance of the multiloop PID controller.

5.2.3.1 Unconstrained MAdGPC with variable forgetting factor

This basic run shows the initialization or the probing sequence for identification that is used to start the adaptive controller as well as the servo-tracking and regulatory capabilities of this unconstrained controller. Before the GPC controller is turned on or set on the automatic mode, the process was excited by giving step changes in both its inputs. The MIMO AUD identification algorithm was used to obtain estimates of the model parameters of the process using the initial data. After allowing the openloop identification to run till the parameters estimates have reached in the range of their respective value, which in other words would mean that the parameter estimates have converged to their respective values and that a sufficiently good model of the process is available, the GPC controller was turned on and then its various properties were analyzed

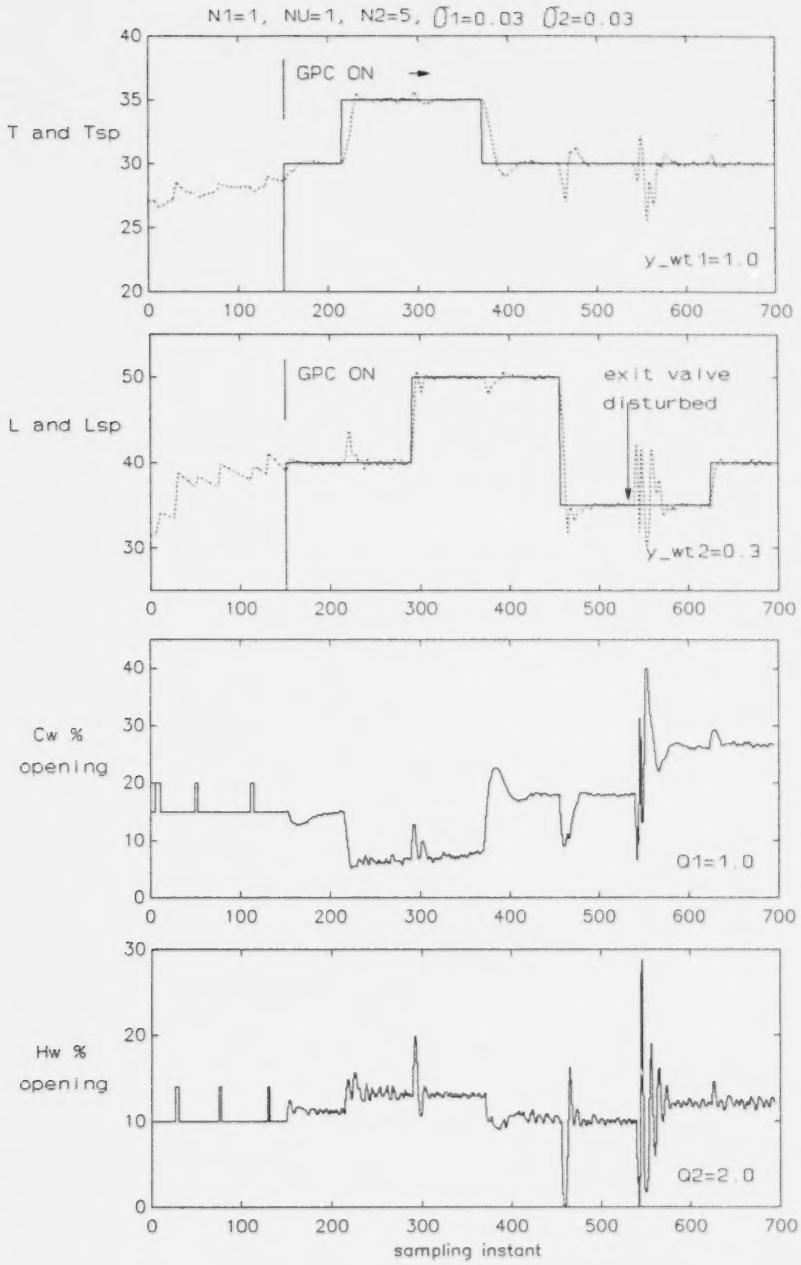


Figure 5.4 Unconstrained MAdGPC with variable forgetting factor including the initialization of the adaptive controller

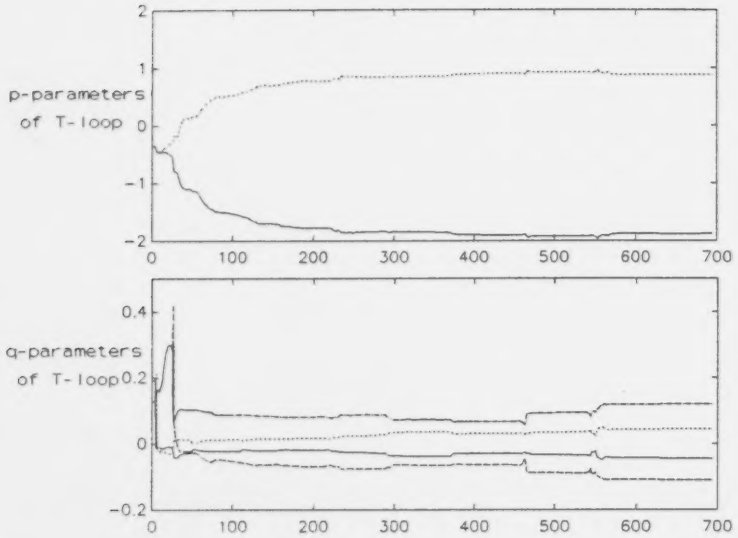


Figure 5.5 Parameter trajectories of the MISO subsystem corresponding to the temperature loop in unconstrained MAdGPC.

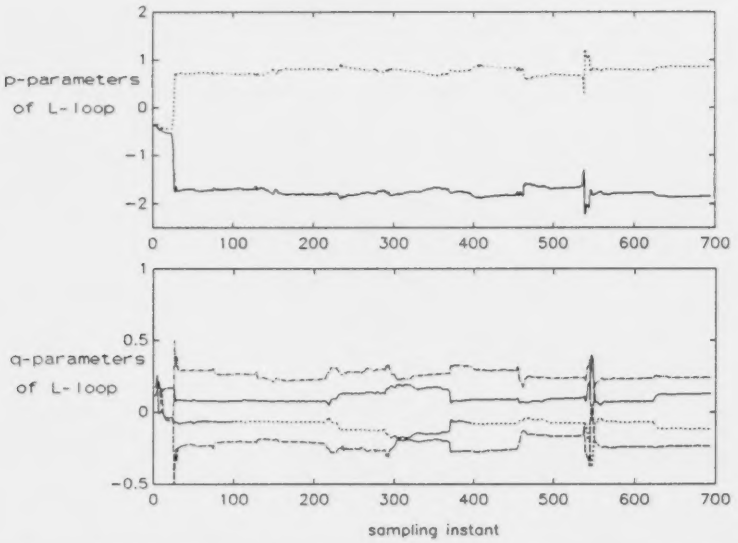


Figure 5.6 Parameter trajectories of the MISO subsystem corresponding to the level loop in unconstrained MAdGPC

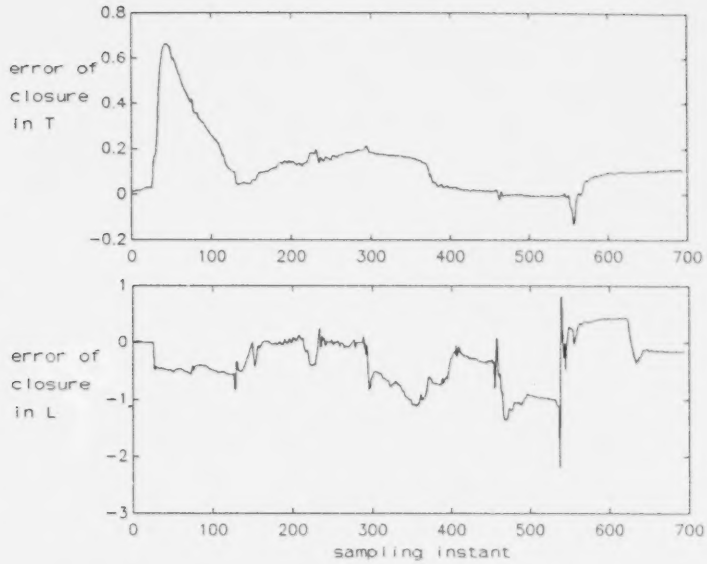


Figure 5.7 Estimates of the error of closure in temperature and level in ARMAX representation in unconstrained MAdGPC.

This procedure was followed for all the experimental runs. The excitation or probing sequence is shown figure 5.4. The excitation signal consisted of step changes of 15 and 10 % valve opening to the cold water and the hot water valves respectively. At sampling instant # 150, the MIMO GPC controller was turned on with the MIMO AUD identification algorithm continued to identify the process parameters at every sampling instant thereafter. The basic servo tracking properties of the adaptive controller are shown in figure 5.4. The deviations from the setpoints of the level trajectory at sampling instant # 220 and # 380 are due to the interaction from the temperature loop due to set-point change in the temperature. The temperature to level interaction can be seen in the temperature graph at sampling instants # 290 and # 460. Inspite of strong coupling between the two loops, this run demonstrates the ability of GPC to handle interactions

in a multivariable system with little difficulty. Also the regulatory or disturbance rejection capabilities are shown in figure 5.4. At sampling instant # 540, a disturbance in the level loop was introduced by shuting and reopening the exit water valve. The controller is able to reject this disturbance and is able to provide good control. Figures 5.5 and 5.6 shows the parameter trajectories of the process. The p-parameters of T-loop and L-loop capture the dynamics of the respective loops. It can be seen that the roots of the p-parameter polynomial of the level loop are very near to the unit circle. Thus, the system is expected to show ringing behaviour (Seborg, Edgar and Mellichamp, 1989), which may lead to marginal instability. Figure 5.7 shows the variation of the parameter corresponding to the one in data vector. This parameter compensates for the error of closure in identification.

5.2.3.2 Effect of the variable forgetting factor on unconstrained MAdGPC

To demonstrate the effect on control of the variable forgetting factor used in the identification part of the adaptive controller, a run was conducted with the following tuning parameters:

$N_1=1$, NU=1, $N_2=5$, $Q_1=1.0$, $Q_2=2.0$, y_wt1=1.0, y_wt2=0.3.

There are two parameters which affect the value of the forgetting factor at each instant, they are the lower bound on the forgetting factor (λ_{lb}) and the *sigma parameter* which is the measure of the variance of the measurement noise (σ). As shown in figure 5.8, initially the lower bound on the forgetting factor were kept at $\lambda_{lb1}=0.95$, $\lambda_{lb2}=0.95$ and the sigma parameter were chosen to be $\sigma_1=0.007$, $\sigma_2=0.01$. At sampling instant 100,

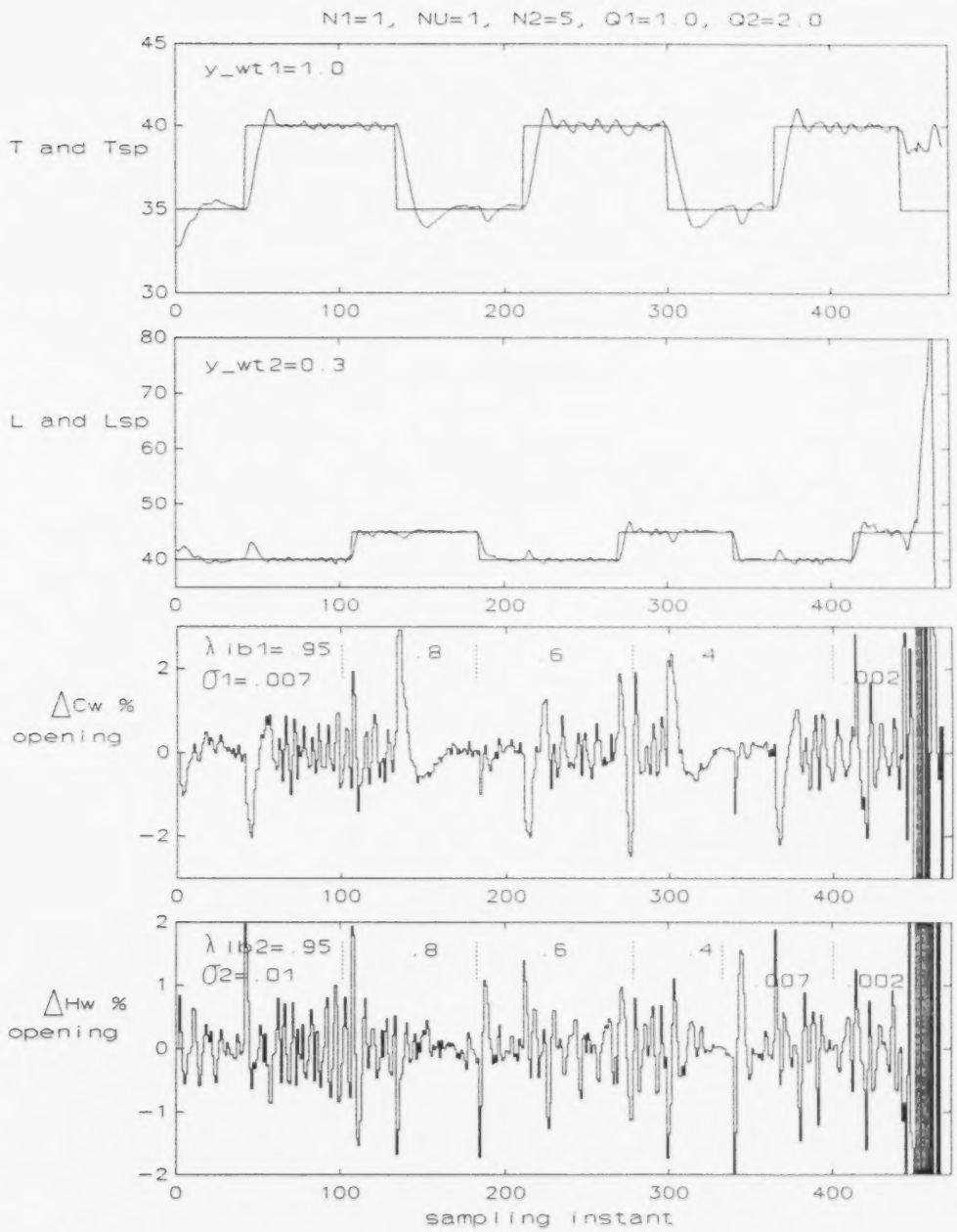


Figure 5.8 Effect of a lower bound on forgetting factor and the sigma parameter on unconstrained MAdGPC.

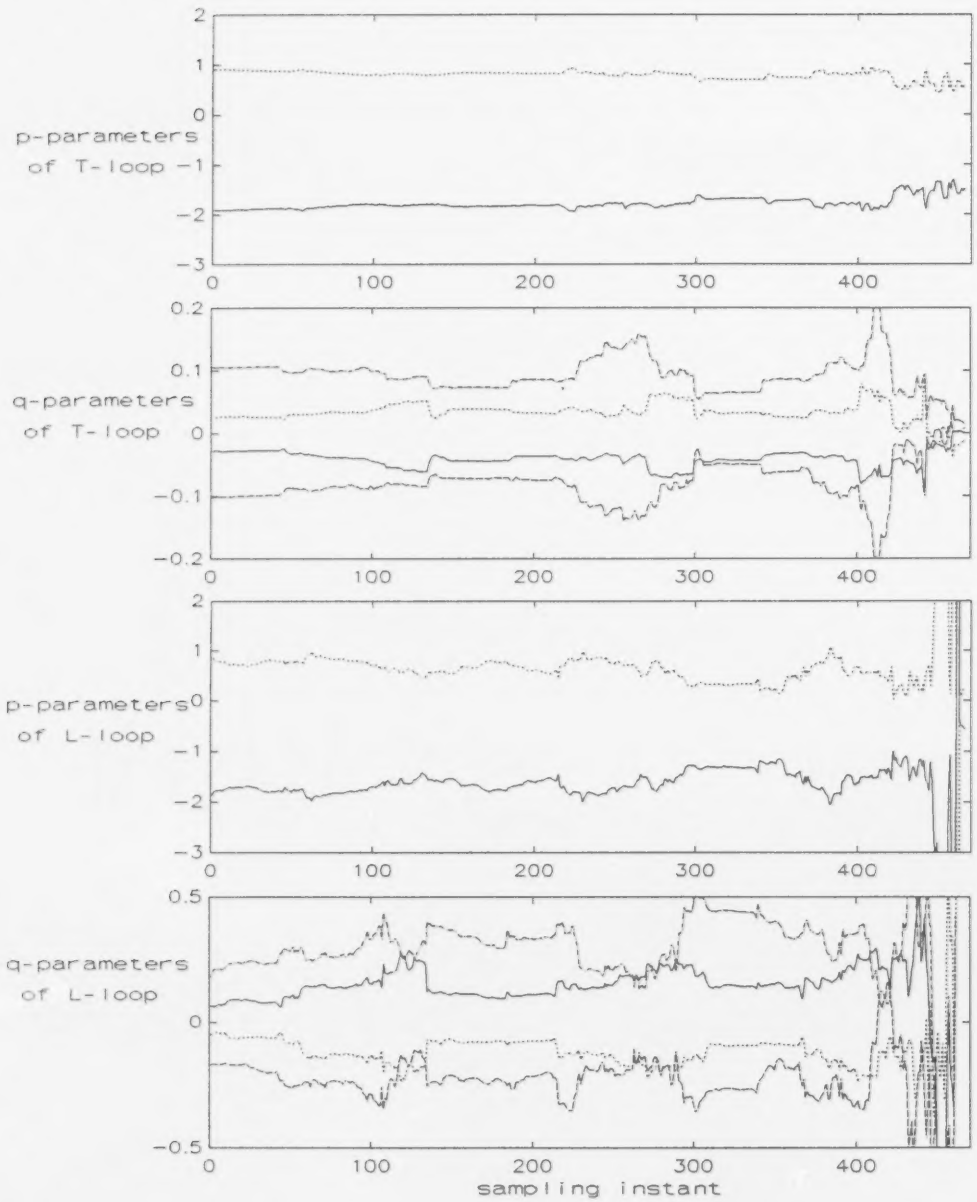


Figure 5.9 Parameter trajectories of the MISO subsystems corresponding to the temperature and level loops due to the variation of a lower bound on forgetting factor value and the sigma parameter.

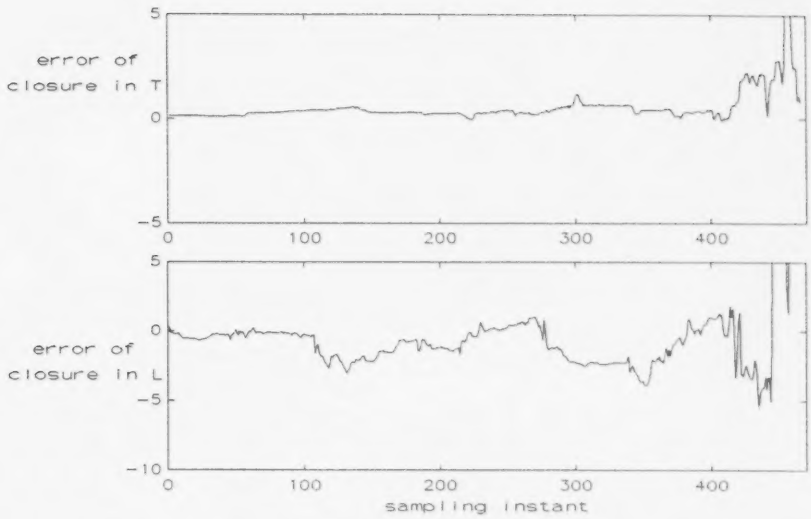


Figure 5.10 Variation in the estimates of the error of closure in temperature and level in ARMAX representation due to the variation in lower bound on forgetting factor and the sigma parameter.

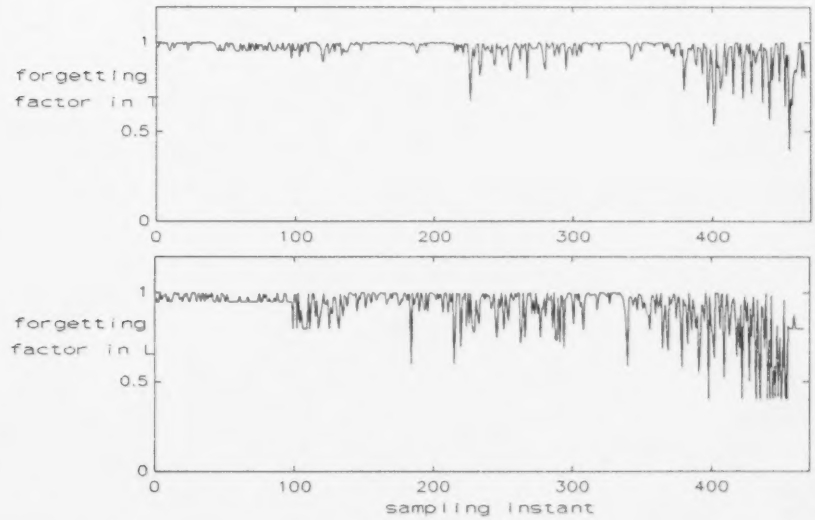


Figure 5.11 Variation in the forgetting factor value due to a lower bound on its value and the sigma parameter.

180 and 275, the lower bounds on the forgetting factor were changed to 0.8, 0.6 and 0.4

respectively. Also shown in the figure are the incremental control action for the cold water and hot water openings. Figure 5.9 shows the parameter estimates. It shows that with decrease in the lower bound on the forgetting factor, the variance in the parameter estimates is higher and consequently the variance in the control action is also higher. This is expected since when the forgetting factor lower bounds are reset to lower values, the identification algorithm is more alert and therefore more sensitive to noise effects and hence the variance of the parameter estimates is higher.

The sigma parameter is used in the calculation of the forgetting factor. At sampling instant 340, the value of the sigma parameter for the level loop is decreased to 0.007 and at sampling instant 400, its value for both loops is changed to $\sigma_1=0.002$ and $\sigma_2=0.002$. Figure 5.9 shows a dramatic increase in the variance of the parameter estimates and consequently the control action after instant 400 which leads to an unstable response. This is because for a fixed low value of lower bound on the forgetting factor, with low sigma parameter, the forgetting factor values become very low, which makes the algorithm very alert and hence sensitive to noise and smallest disturbance in the process. The parameters vary a lot from their actual values which gives a rapidly changing, poor model of the process, which will lead to oscillatory control action and perhaps an unstable response as shown in figure 5.8 and 5.9. Thus selection of the sigma parameter can be very important and a wrong choice of which may lead to an unstable response. It is not advisable to use very low values of σ. The value is specific to the process, thus a safe choice should be made such that the forgetting factor values do not become very low.

5.2.3.3 Effect of Maximum Prediction Horizon (N_2)

This run shows the effect of finite-time prediction horizon parameter N_2 on the performance of the adaptive controller. It can be seen from figure 5.12, that for constant values of $N_1=1$, $NU=1$ and $N_2=2$ until sampling instant # 80, the response is very sluggish. When N_2 is increased to 4 at sampling instant # 80 and to 6 at sampling instant # 220, the response is faster. However as N_2 is increased further, for example when N_2 is made equal to 8 at sampling instant # 320 the control actions becomes oscillatory and aggressive as can be seen from figure 5.12. When N_2 was further increased to 10 at sampling instant # 420, the response is very aggressive and leads to an almost instability. The process is brought back to a more satisfactory control by setting $N_2=5$ at sampling instant # 520. Mohtadi (1986) and Clarke *et al.*(1989) have reported that when N_2 approaches a relatively large number and with $NU=1$, then GPC behaves like a mean-level controller. For a mean level controller, the closed loop poles remain in their open loop positions (McIntosh *et al.*, 1991). For this process (see section 5.2.3.1) the open loop poles of the process are very near the unit circle. With mean level control, the closed loop poles would also be expected to be near the unit circle thus resulting in an oscillatory performance at larger values of N_2. The selection of N_2 is therefore an important tuning parameter of the MAdGPC algorithm.

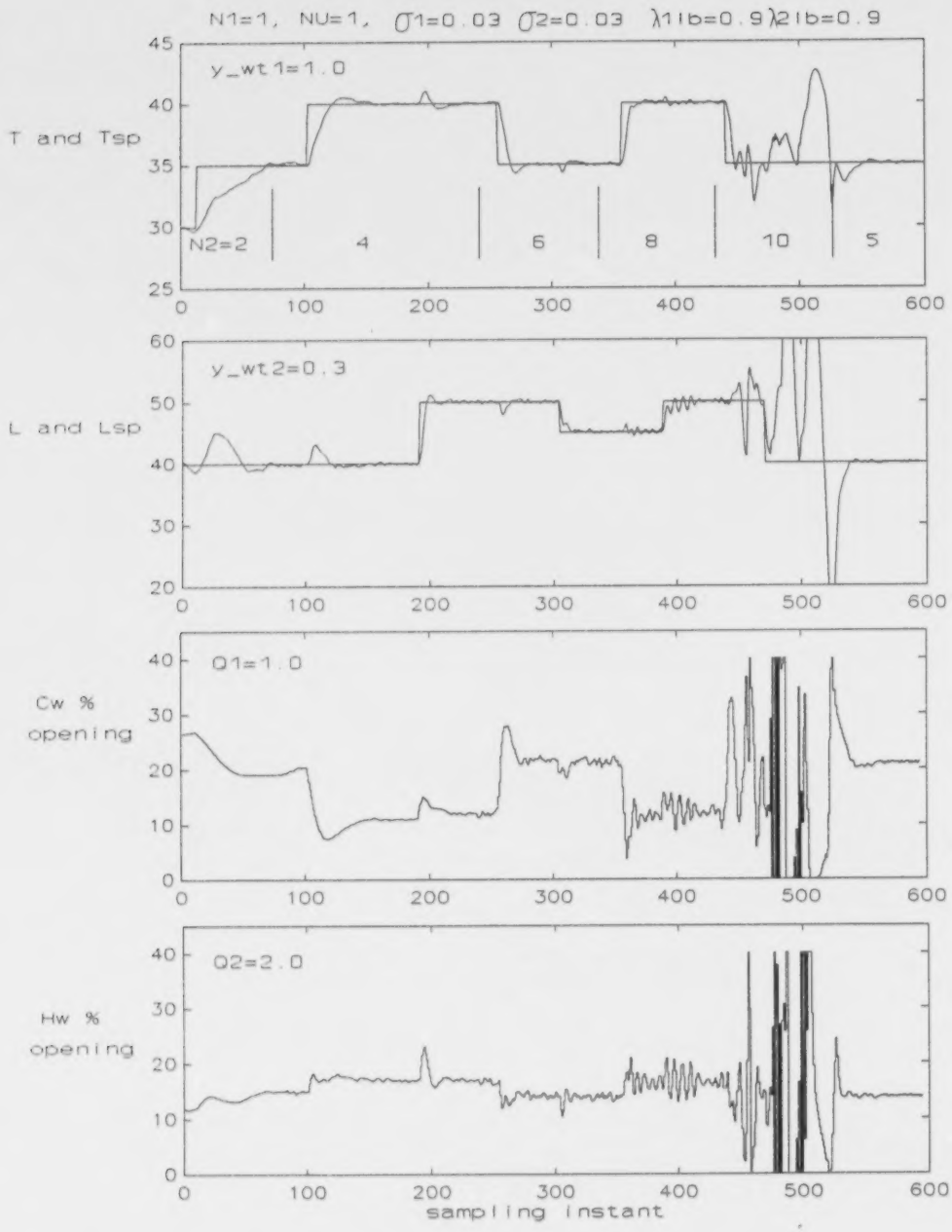


Figure 5.12 Effect of maximum prediction horizon (N_2) on the performance of Unconstrained MAdGPC.

5.2.3.4 Effect of Control Input Weighting

This run emphasizes the importance of the control input weighting terms on the performance of the GPC controller. In this run the values of the output weighting terms were kept constant, and the effects of the variation of input weighting were studied. The output weighting values were chosen to be y_wt1=1.0 and y_wt2=0.3. The values of other important tuning parameters were $N_1=1$, NU=1, $N_2=5$.

The input weighting terms are the coefficients of the inputs in the cost function, and they serve to emphasize the cost of 'energy'. If the value of the input weighting is chosen to be very small, then that input is considered to be less expensive and large excursions of this variable can be used at relatively smaller cost to minimize the error between the output and its set-point. This will result in very aggressive control action and can often lead to oscillatory output responses. On the other hand if the value of the input-weighting is chosen to be very large, then that input is considered to be relatively expensive and its usage will have to be very conservative. This will give a stable but slower control action leading to a very sluggish output response.

The run shown in figure 5.13, starts with the input weightings set to $Q_1=0.5$ and $Q_2=0.5$. It gives stable output responses. At sampling instants 90 and 140, the input weighting values were decreased to $Q_1=0.2$, $Q_2=0.2$ and to $Q_1=0.1$, $Q_2=0.1$ respectively. The result is an aggressive control policy leading to unstable output responses. Also the variance in the parameter estimates as seen in figures 5.14 and 5.15 is very high, which will further deteriorate the performance of the controller. To stabilize the output response, the input weighting values were brought back to $Q_1=0.5$ and

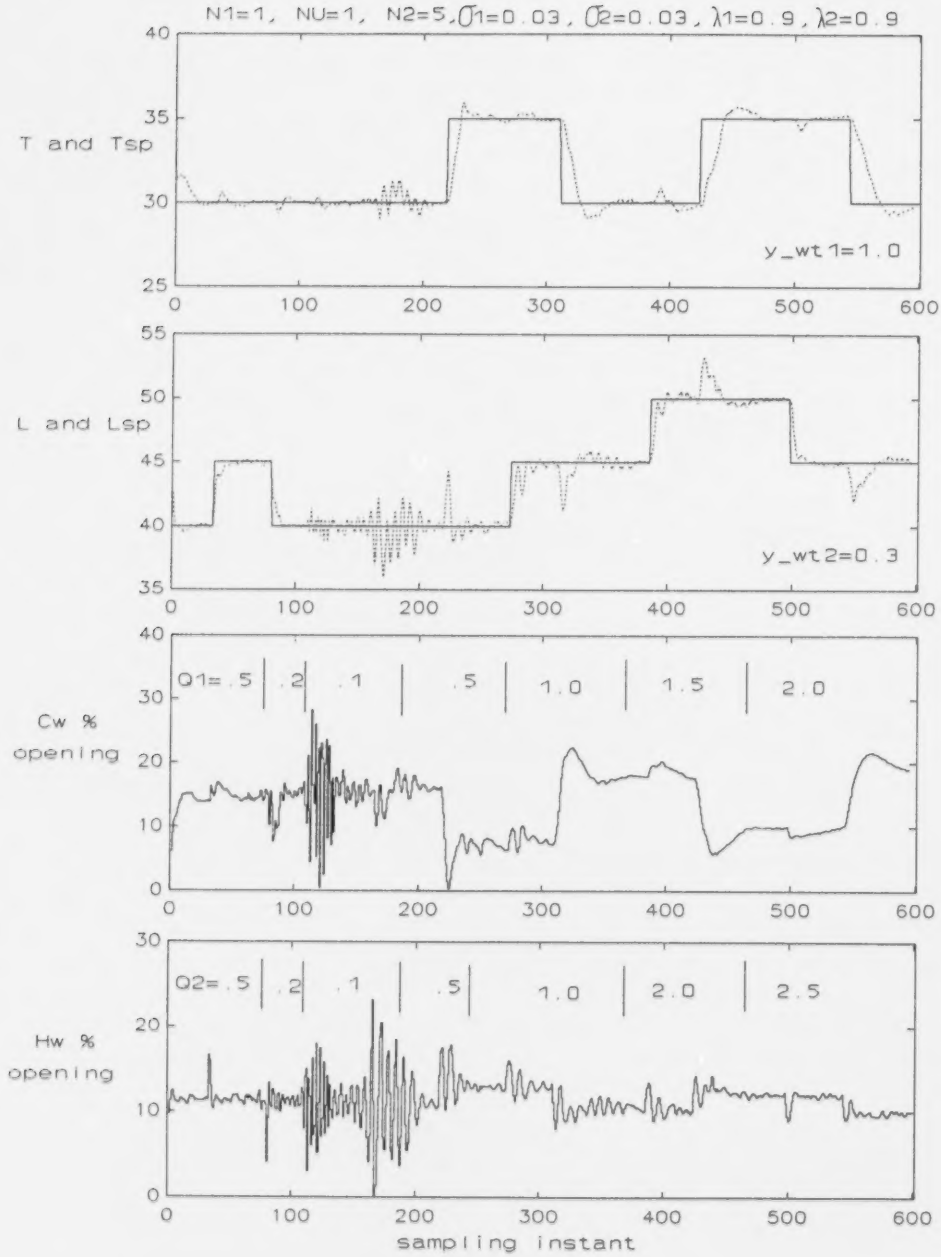


Figure 5.13 Effect of input weighting (Q) on the performance of unconstrained MAdGPC.

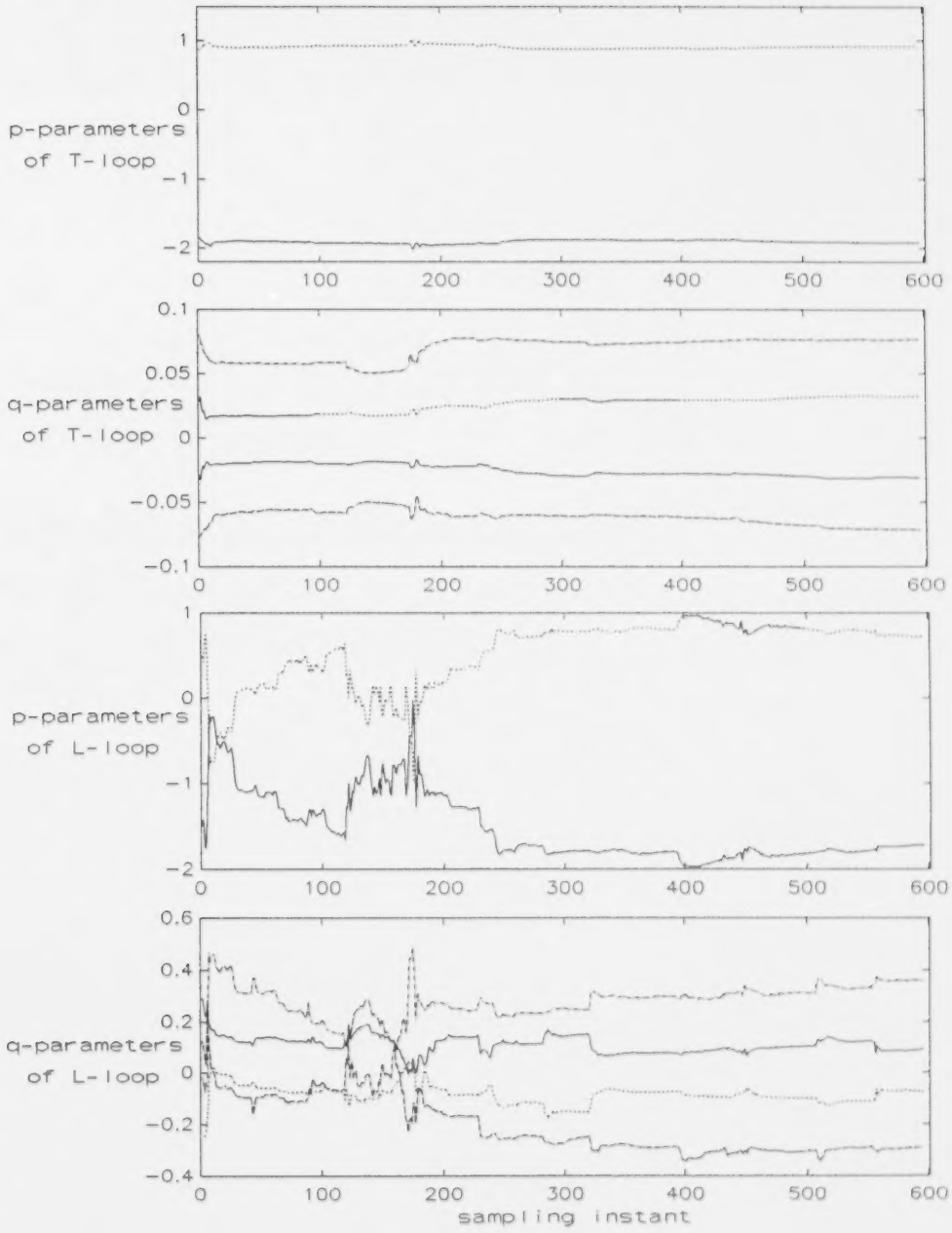


Figure 5.14 Parameter trajectories of the MISO subsystems corresponding to the temperature and level loops due to variation in the input weighting.

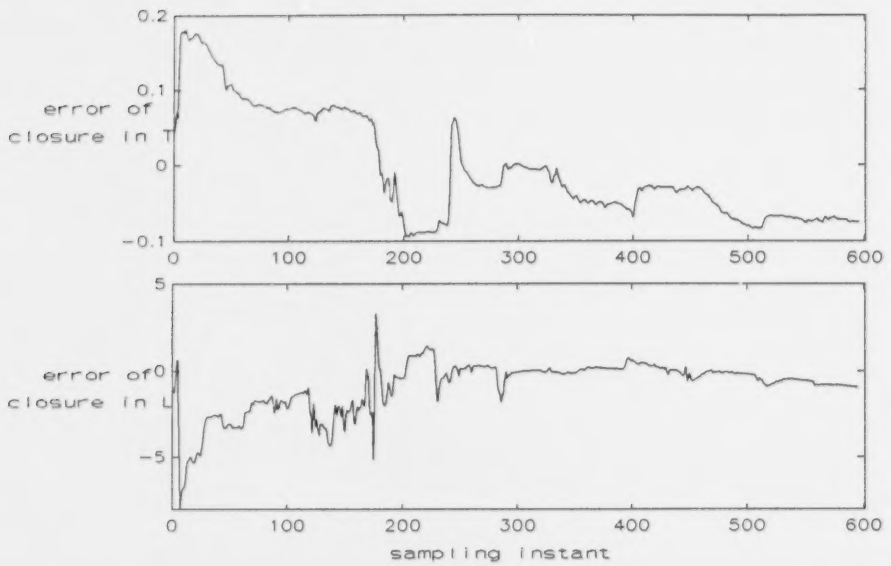


Figure 5.15 Variation in the estimates of the error of closure in temperature and level in ARMAX representation due to the variation of input weighting.

Q_2=0.5 at sampling instant # 180. Since the second input was continuing to oscillate, the input weighting on the second input was increased to Q_2=1.0 at sampling instant 230. The responses were further smoothened by increasing Q_1=1.0 at sampling instant 260. The resulting control actions are less aggressive leading to a stable response. The input weightings are further increased to Q_1=1.5, Q_2=2.0 and Q_1=2.0, Q_2=2.5 at sampling instants 360 and 470 respectively. It can be seen from figure 5.13 that the output response becomes sluggish with increase in the input weightings. The input control weighting parameter allows a trade-off between stability and speed of the output response. Thus the value of this tuning parameter is very important to get a stable, desirable performance of the controller.

5.2.3.5 Effect of Output weighting

This section evaluates the significance of output weighting parameters on the MAdGPC performance. This parameter is of relevance only in MIMO controllers. The effect of output weights is studied here by keeping the input weights at a constant value of $Q_1=1.0$ and $Q_2=2.0$ with remaining tuning parameters set at $N_1=1$, $NU=1$, $N_2=5$. It is important to remember that it is the ratio of the output weightings and not their absolute value that is important. Thus the output weighting parameter for temperature (y_wt1) was fixed at 1 and the output weighting for level (y_wt2) was varied.

Initially the level is given very little weight relative to the temperature by setting y_wt2=0.01. The level will not track its set-point since only 1% of the control action is used to minimize the error in level. The weight on the level (y_wt2) is increased to 0.1 and 0.3 at sampling instants 80 and 210 respectively. The result is faster response of level to a setpoint change. At sampling instants 320, 410 and 460, y_wt2 is increased to 0.6, 0.8 and 1.0 respectively, which leads to more ringing in the control action consequently making the level response oscillatory and the temperature response is also becoming oscillatory. This is due to a large time constant of the system, which may cause ringing in the process under closed-loop control which was discussed in section 5.3.2.1. The output weighting parameters emphasize the relative importance of each loop from the viewpoint of control. If the level loop has a smaller weight associated with it relative to the temperature loop, then this will force the control algorithm to regulate the temperature loop more tightly compared to the level loop. Improper selection of output weights may lead to an unstable system.

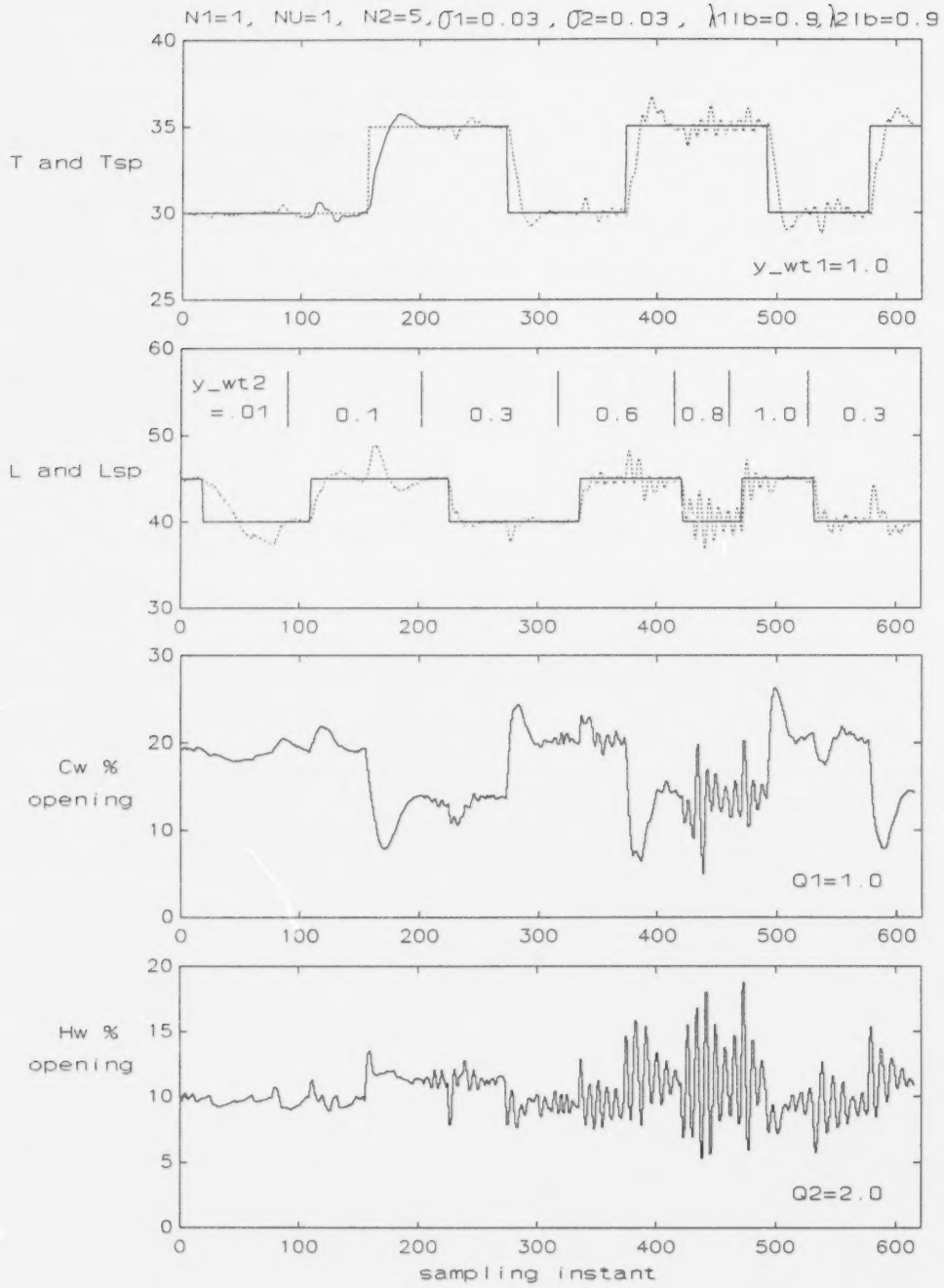


Figure 5.16 Effect of output weighting on the performance of unconstrained MAdGPC.

5.2.3.6 Effect of T-filter

This is a disturbance tailoring filter which helps in handling high frequency disturbances in the process, which may be due to the unmodelled dynamics or the presence of unmeasurable load disturbances. To demonstrate the effect of a T-filter the other tuning parameters of the controller are fixed at:

$N_1=1$, NU=2, $N_2=5$, y_wt1=1.0, y_wt2=0.3, $Q_1=1.5$, $Q_2=2.5$.

Initially, the T-filter is $[1 - 0.8q^{-1}]$, as recommended by McIntosh (1988). It can be seen from figure 5.17, that the controller for the above chosen tuning parameters gives a smooth and stable response. At sampling instant # 150, when the T-filter is turned off, the response becomes fairly oscillatory and slowly damps out tending to stability after sampling instant # 200. The T-filter was turned on again at sampling instant # 220, and a very stable and smooth response as obtained initially was reproduced. This run illustrates that the T-filter is an important tuning parameter for satisfactory performance of MAdGPC.

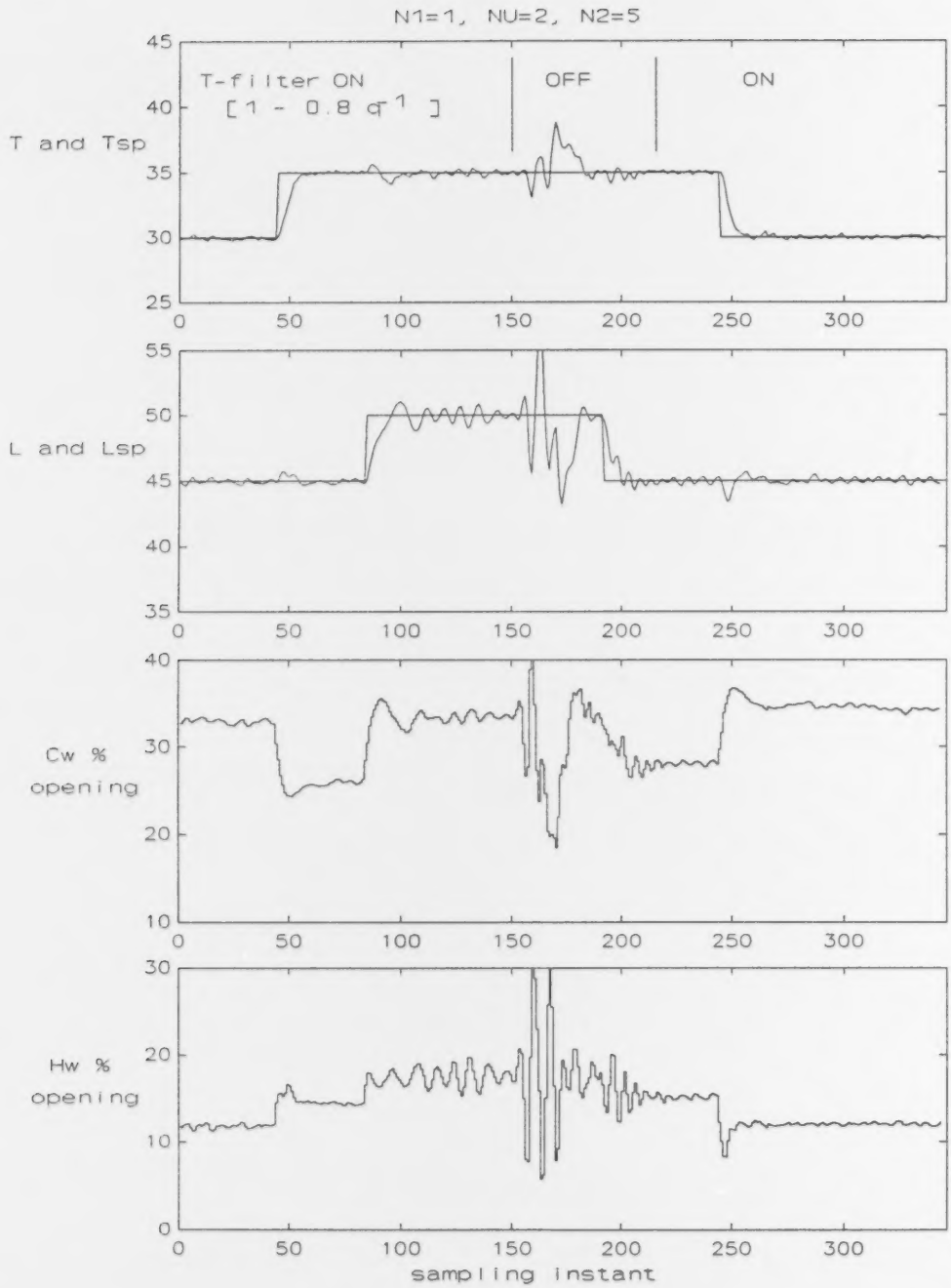


Figure 5.17 Performance of unconstrained MAdGPC with and without the T-filter.

5.2.3.7 Effect of Input Rate Constraints

This run compares the performance of the adaptive controller in the presence of constraints on the rate of input control action. These constraints are enforced by the optimization scheme using a commercial optimization package QPSOL from the Stanford University Optimization laboratory (Gill *et al.*,1984).

In the first constrained control run no bounds on the amplitude of the input have been placed. The controller was intentionally poorly tuned to illustrate the effect of rate constraints. The tuning parameters values chosen were $N_1=1$, $NU=1$, $N_2=8$. The input and the output weighting were fixed at $Q_1=1.0$, $Q_2=2.0$ and $y_wt1=1.0$, $y_wt2=0.3$. The lower bounds on the forgetting factor values were chosen to be $\lambda_{1lb}=0.9$ and $\lambda_{2lb}=0.9$. As shown in figure 5.18, initially the rate constraints imposed on both the inputs of the controller were, $\Delta U_{max}=+2$ and $\Delta U_{min}=-2$. Since the controller is poorly tuned the performance is not very good. Nevertheless, the response is stable. At sampling instant # 180, the constraints on the rate of input action were loosened and the values used are, $\Delta U_{max}=+20$ and $\Delta U_{min}=-20$. The control action is very oscillatory as seen from figure 5.18 which gives oscillatory response. The performance of the controller is improved by one again resetting the rate constraints to their original values at sampling instant 225. By imposing rate constraints, ringing which would occur in an open loop marginally unstable process like the one discussed here can be reduced considerably, but this will give a sluggish response.

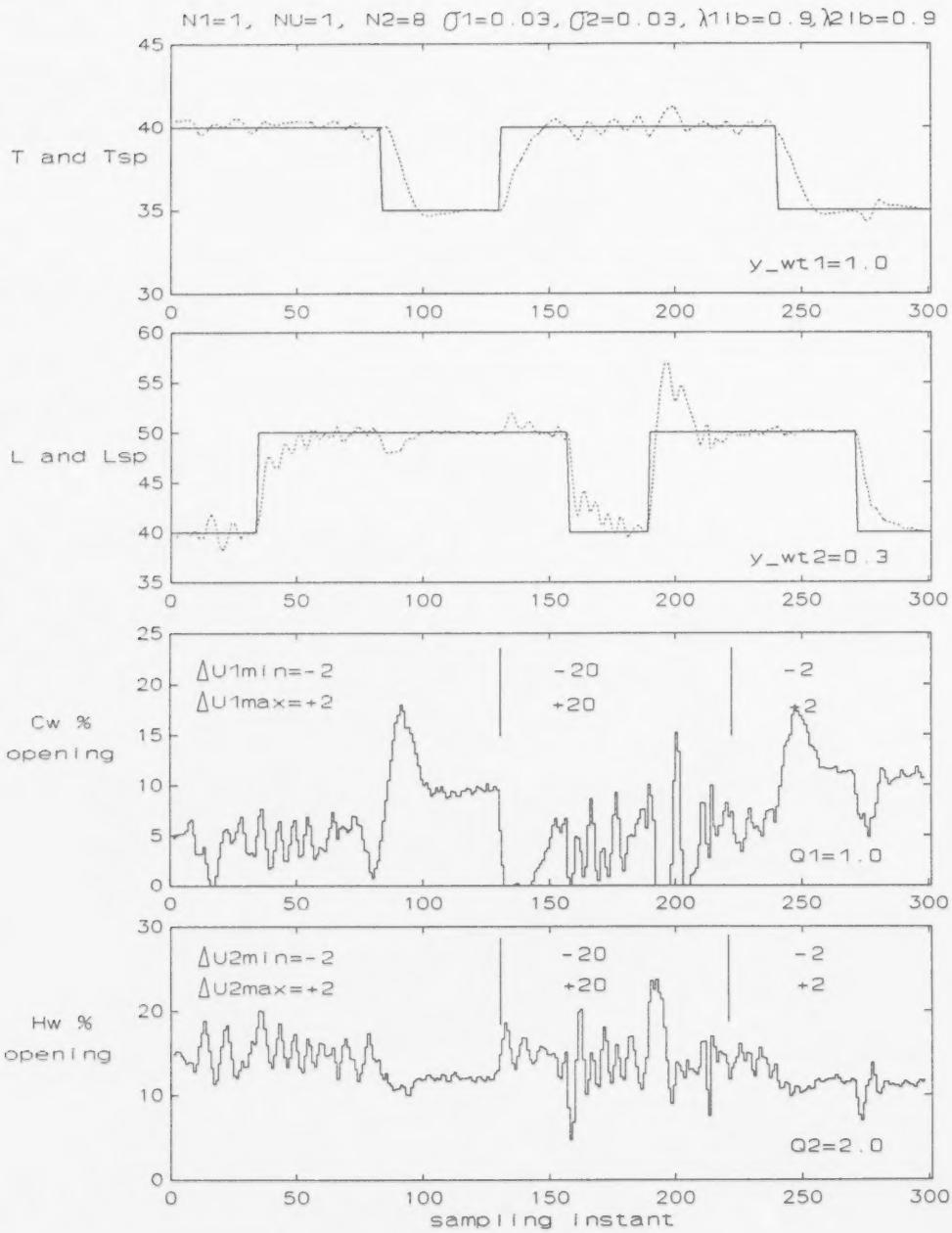


Figure 5.18 Performance of MAdGPC in the presence of rate constraints on inputs solved using QPSOL optimization package.

5.2.3.8 Effect of Input Amplitude Constraints

This run illustrates the effect of amplitude constraints on the inputs on the control algorithm and how the choice of the upper and lower bounds on the values of the control action can affect the performance of the controller.

The rate constraints were removed for this run to study the effect of amplitude constraints on MAdGPC. The tuning parameters chosen were, $N_1=1$, $NU=1$, $N_2=5$, $Q_1=1.0$, $Q_2=2.0$, y_wt1 = 1.0, y_wt2 = 0.3. Initially the amplitude constraints were fixed at $U_{1max}=20$, $U_{1min}=10$ and $U_{2max}=13$, $U_{2min}=10$. It can be seen from figure 5.19, that at sampling instant 70, the value of U_1 is saturated at 10, its lower bound and at sampling instant 80, U_2 is saturated at 13, its upper bound. The temperature is below its set-point and the level is above its set-point and these two outputs variables are unable to go to their respective set-point values, because of the saturation of the inputs. At sampling instant 95, when the lower constraint on U_1 is released to 5, the input action decreases and both the outputs were able to reach their respective set-point values. This experiment is repeated at sampling instant # 170 where U_{1min} was released to 0.

This run has served to demonstrate that the values of the bounds selected affect the performance of the controller. These constraints are usually imposed to force the controller to calculate only physically realizable signals, for example, a control valve cannot open below 0 % and above 100 %, thus the upper and lower constraints on the amplitude of the control input can be selected as 100 and 0 respectively.

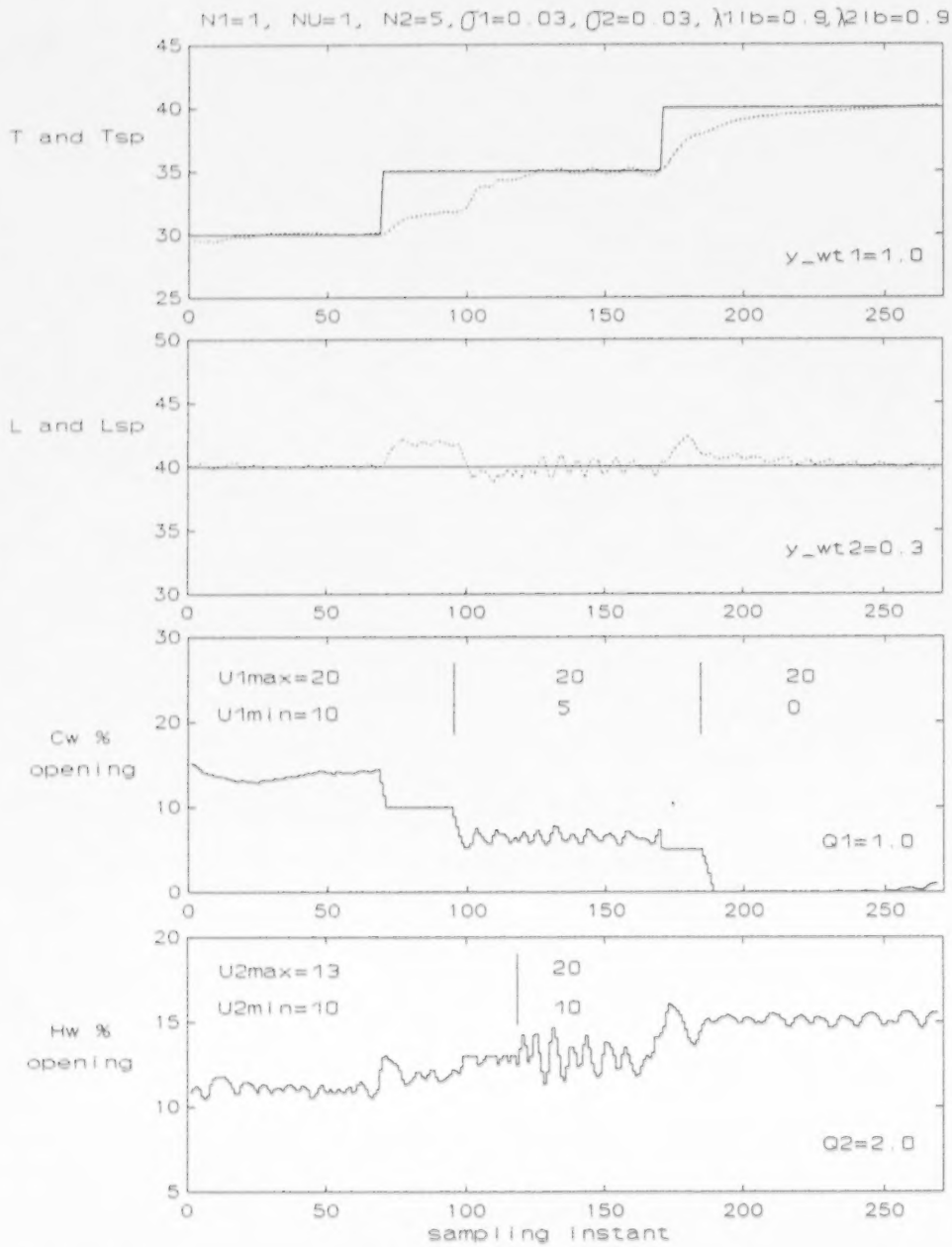


Figure 5.19 Performance of MAdGPC in the presence of amplitude constraints on inputs solved using the QPSOL optimization package.

5.2.3.9 Analytical Input Amplitude and Rate Constraints on MAdGPC

This run studies the analytical solution for rate and amplitude constrained MAdGPC for special case of NU=1 on CSTH. The algorithm was implemented using the MIMO AUD algorithm with constrained non-adaptive MIMO GPC. The analytical solution of the constrained GPC is realized using Kuhn-Tucker Multipliers Method, which has been developed by Mutha (1990).

To study the rate constrained MAdGPC, the amplitude constraints were removed and the tuning parameters were fixed at $N_1=1$, NU=1, $N_2=8$, which were intentionally poorly tuned values to study the effect of rate constraints. Initially, the rate constraints imposed on both the inputs were $\Delta U_{min}=-1$ and $\Delta U_{max}=+1$. It can be seen from figure 5.20, that the response is stable but very sluggish. At sampling instant # 150, the rate constraints were relaxed to $\Delta U_{min}=-5$ and $\Delta U_{max}=+5$, the resulting control action becomes very aggressive and leads to an almost unstable response. At sampling instant # 280, the tighter constraint values of $\Delta U_{min}=-1$ and $\Delta U_{max}=+1$ were imposed and the initial stable performance was reproduced. Figure 5.20 demonstrates that the rate constraints are operational and that analytical solutions do work. The effect of rate constraints via an analytical solution during regulatory performance is illustrated in figure 5.21. Notice that the disturbances are countered with larger magnitudes and incremental changes in the control action when large limits are enforced.

To study the amplitude constrained MAdGPC, the rate constraints were removed and the tuning parameters were fixed at $N_1=1$, NU=1, $N_2=5$. Initially, the amplitude constraints on both the inputs were $U_{min}=10$ and $U_{max}=20$. It can be seen in figure 5.22, that the

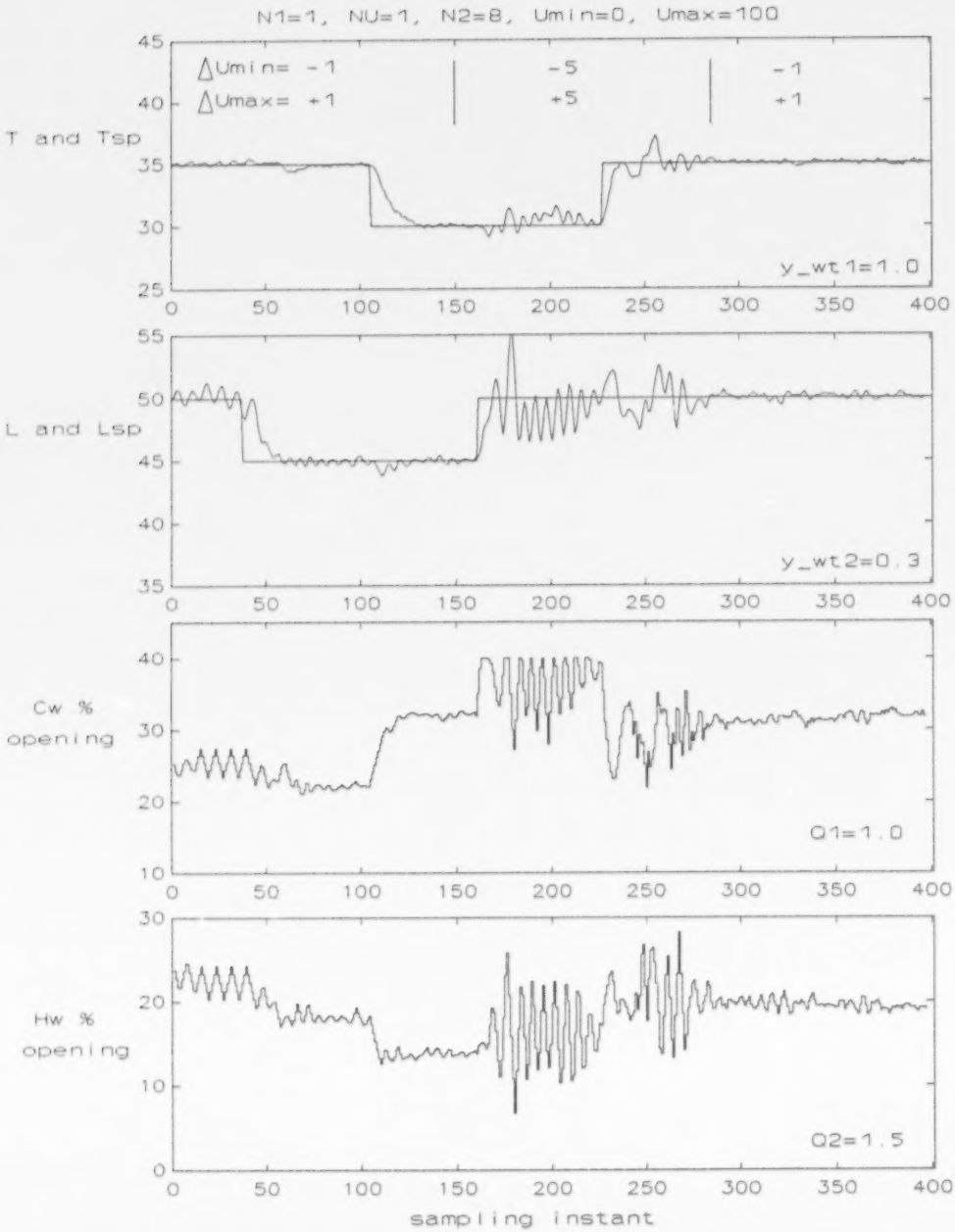


Figure 5.20 Performance of MAdGPC in the presence of rate constraints on inputs with analytical solution of the constraints.

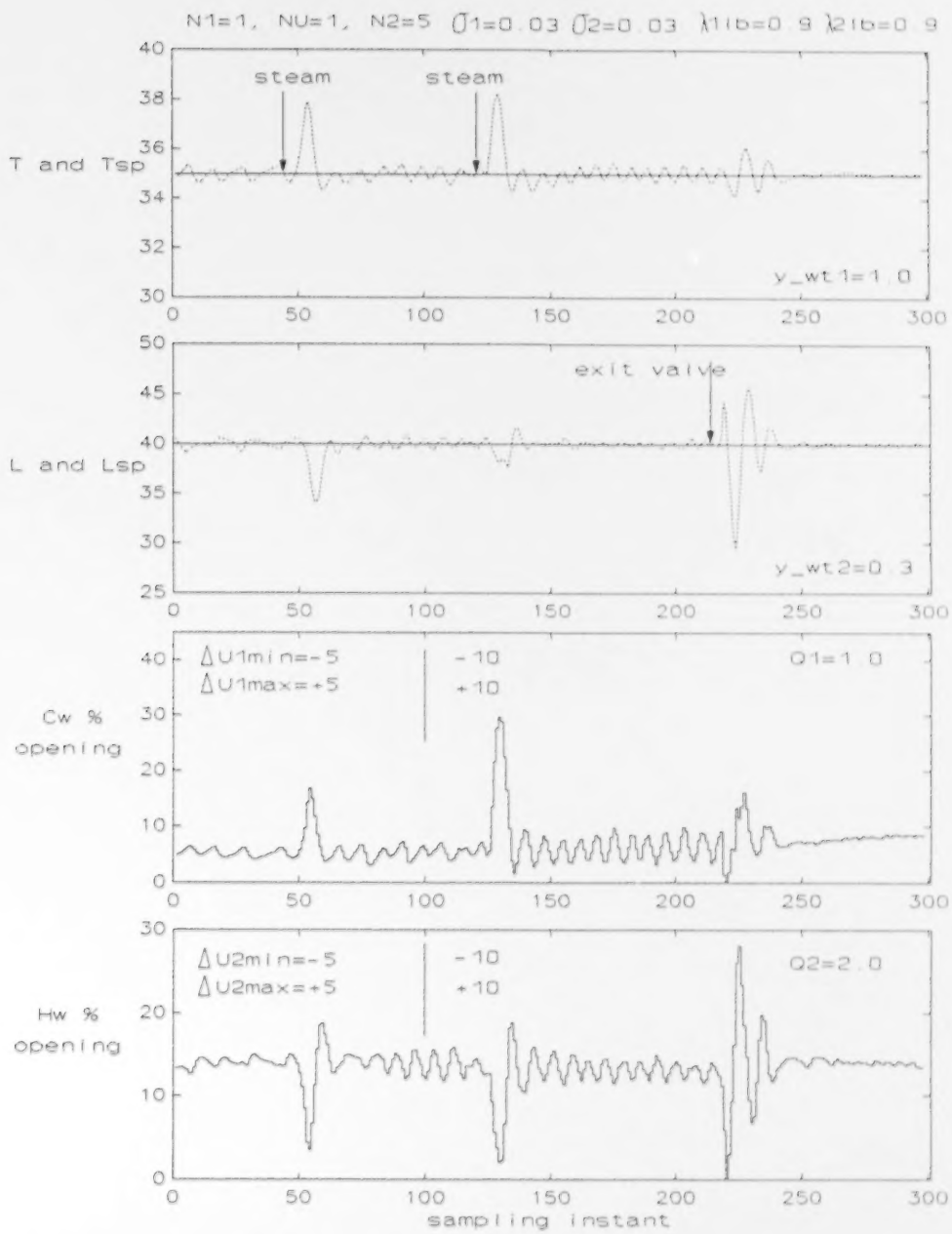


Figure 5.21 Performance of MAdGPC with analytical solution of the rate constraints on inputs in presence of disturbances.

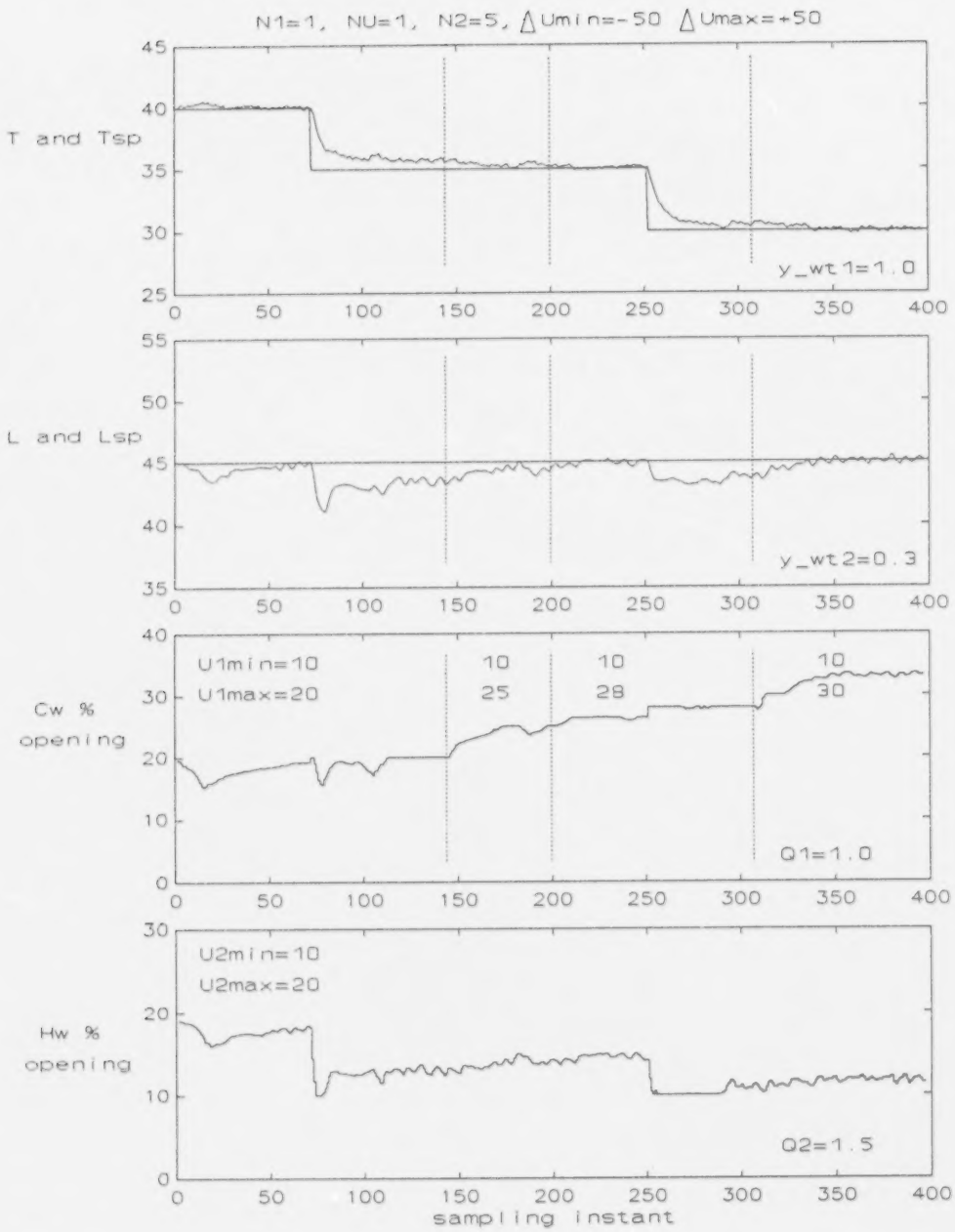


Figure 5.22 Performance of MAdGPC in the presence of amplitude constraints on inputs with analytical solution of the constraints.

cold water valve is saturated at 20% valve opening (its upper constraint value) and as a result the level is below its set-point and the temperature remains above its set-point. At sampling instant # 145, when the upper constraint value of cold water valve is changed to U_{1max}=25, both the outputs are able to reach their respective set-points. Similar behaviour is repeated at sampling instant # 200, where U_{1max} is set to 28 and at sampling instant # 310, when the U_{1max} is set to 30. Thus, it can be seen that the amplitude constraints are operational.

5.2.3.10 Effect of output constraints on MAdGPC

Output constraints are very important from the view point of industrial control. In industry there are various situations in which it is important to regulate a particular output between certain limits, for example the temperature in a polymer reactor should not cross a certain limit, since beyond this limit, it will cause a thermal runaway and also it will have harmful effects on the properties of the polymer. In such situations, it is important to have constraints on the values of the output.

This run demonstrates the capability of the controller to handle output constraints and it also shows the effect of constraints on the performance of the controller. These constraints are enforced into the optimization scheme by using a commercial optimization package QPSOL from the Stanford University Optimization laboratory (Gill *et al.*, 1984).

As discussed in chapter 4, the output constraints have been translated into equivalent rate constraints on the inputs and then implemented. Because this is an adaptive algorithm,

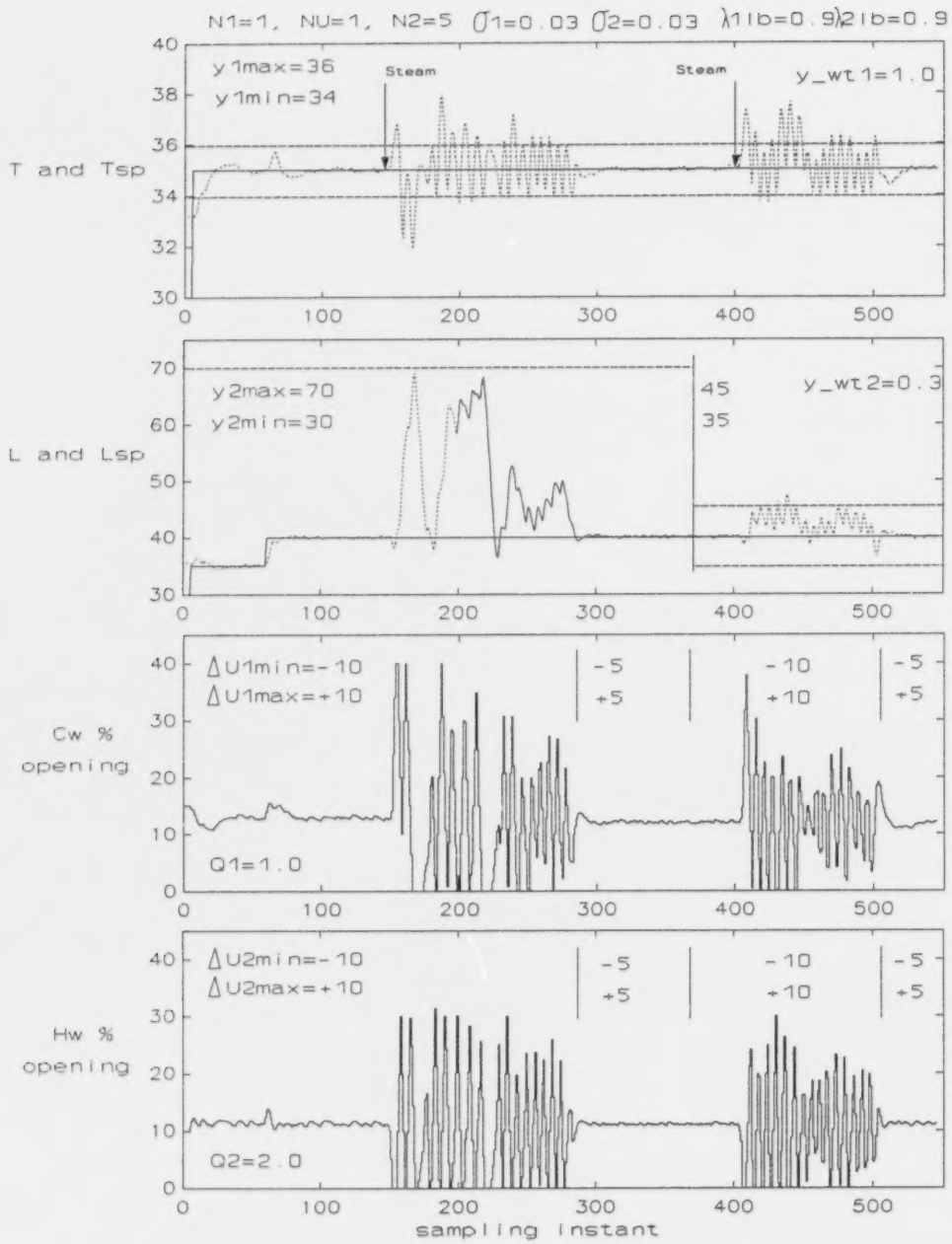


Figure 5.23 Performance of MAdGPC in the presence of output amplitude constraints solved using the QPSOL optimization package.

the model of the process is changing at every time instant and thus the translated constraints on the rate of change of inputs will also change at every time instant. Figure 5.23 shows the performance of the constraints which are solved using QPSOL optimization package. The tuning parameters of the controller are set at:

$N_1=1$, $NU=1$, $N_2=5$, $Q_1=1.0$, $Q_2=2.0$, y_wt1=1.0, y_wt2=0.3, $\sigma_1=0.03$, $\sigma_2=0.03$, $\lambda_{1lb}=0.9$, $\lambda_{2lb}=0.9$.

The rate constraints on both the inputs have been fixed at -10 and +10, the output constraints on temperature are imposed at $y_{1min}=34$, $y_{1max}=36$ and on the level are imposed at $y_{2min}=30$, $y_{2max}=70$. A disturbance in the temperature at sampling instant 150 is introduced by opening the steam valve to 10 % and shuting it after 10 samples. Figure 5.23 shows the effect of the disturbance and the performance of the controller in presence of output constraints. It can be seen that the response of the level is compromised considerably to maintain temperature between the constraint values. It is able to maintain the temperature between the bounds but gives a very oscillatory response. The system is stabilized by tightening rate constraints on the inputs to -5 and +5 at sampling instant 290. Similar performance is repeated by narrowing the constraints on level to $y_{2min}=35$ and $y_{2max}=45$. Notice that output constraints are violated at several instances. Infact QPSOL several times during the sampling instances 150-300 and 400-500, terminated without finding a feasible solution (this occurence is notified to the user by an error message from QPSOL). This is expected in the presence of larger disturbances, i.e. it is impossible to maintain the outputs within the constraint region even with the manipulated variables set at its minimum or maximum limits.

5.2.3.11 The Servo and Regulatory properties of Constrained MIMO Adaptive GPC

This run serves to demonstrate the servo-tracking and regulatory properties of the controller. The complete run can be seen in figure 5.24. The corresponding parameter trajectories are shown in figures 5.25 and 5.26. The controller tuning parameters were set to the values that gave the best performance in earlier runs:

$N_1=1$, $NU=1$, $N_2=5$, y_wt1=1.0, y_wt2=0.3, $Q_1=1.0$, $Q_2=2.0$, $\lambda_{1lb}=0.9$, $\lambda_{2lb}=0.9$.

The amplitude constraints on both the inputs were fixed at $U_{min}=0$ and $U_{max}=100$. The rate constraints on both the inputs were fixed at $\Delta U_{min}=-2$ and $\Delta U_{max}=+2$.

Set-point changes were initiated in both, the temperature and the level loops to study the setpoint tracking properties and the regulatory performance was studied by introducing external disturbances. The interactions are seen at sampling instants 50 and 140 in level due to setpoint change in temperature and at sampling instant 70 in temperature due to a setpoint change in level. The interaction effects are considerably smaller in this constrained controller as compared to the unconstrained controller in section 5.2.3.1. From figure 5.24, it can be seen that the outputs reach their respective setpoint values relatively fast and overall the controller enables the outputs to track their setpoints very well. Thus, the MAdGPC gives a good and satisfactory performance obeying the imposed constraints for these tuning parameter values.

To study the regulatory properties of the controller, various disturbances were given to the process. Disturbance to the temperature in the form of 10% steam valve opening for 20 samples and 10 samples were introduced at sampling instants 220 and 400 respectively. It can be seen that the controller was able to reject these disturbances

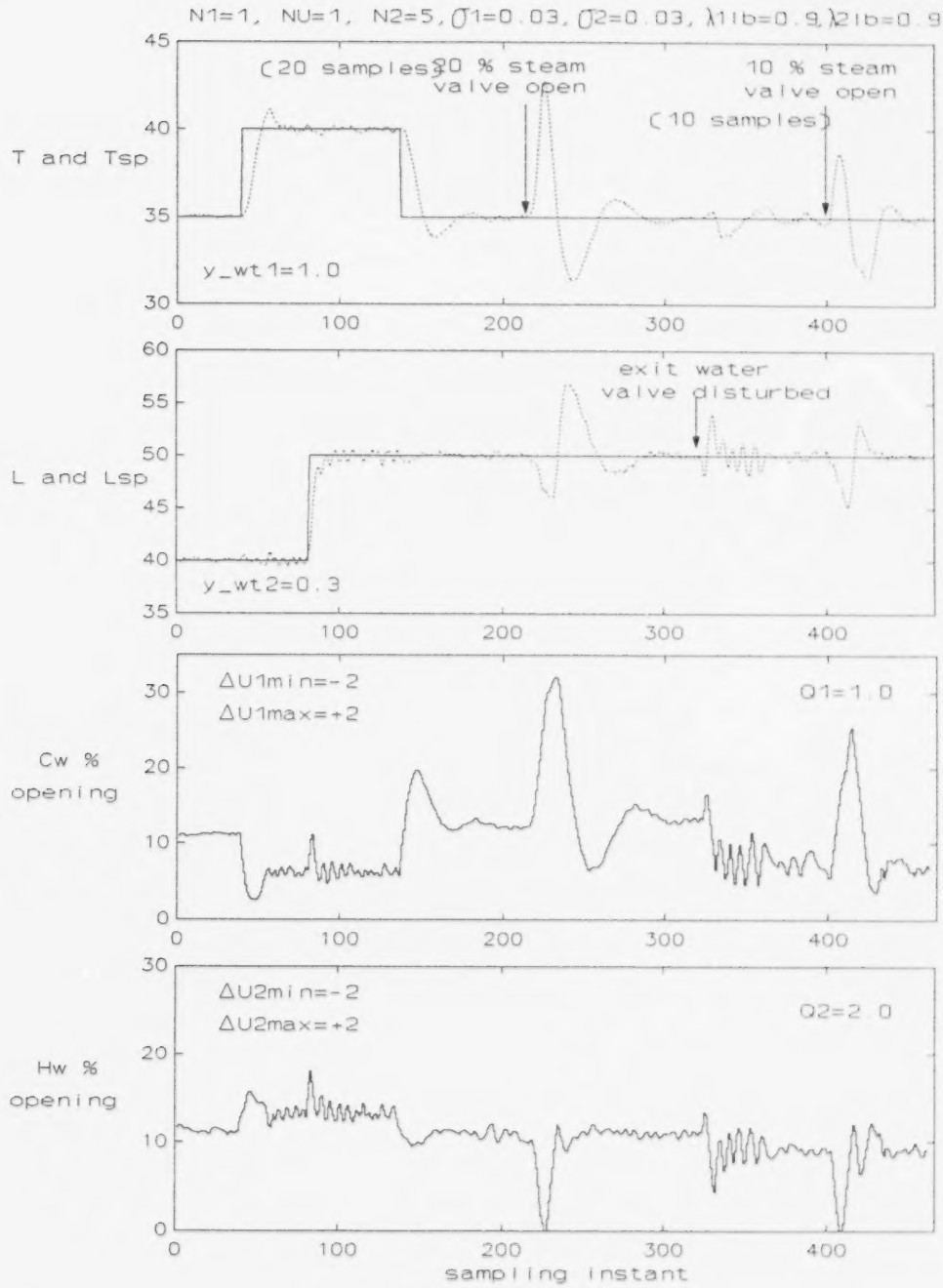


Figure 5.24 Servo and regulatory properties of Constrained MAdGPC.

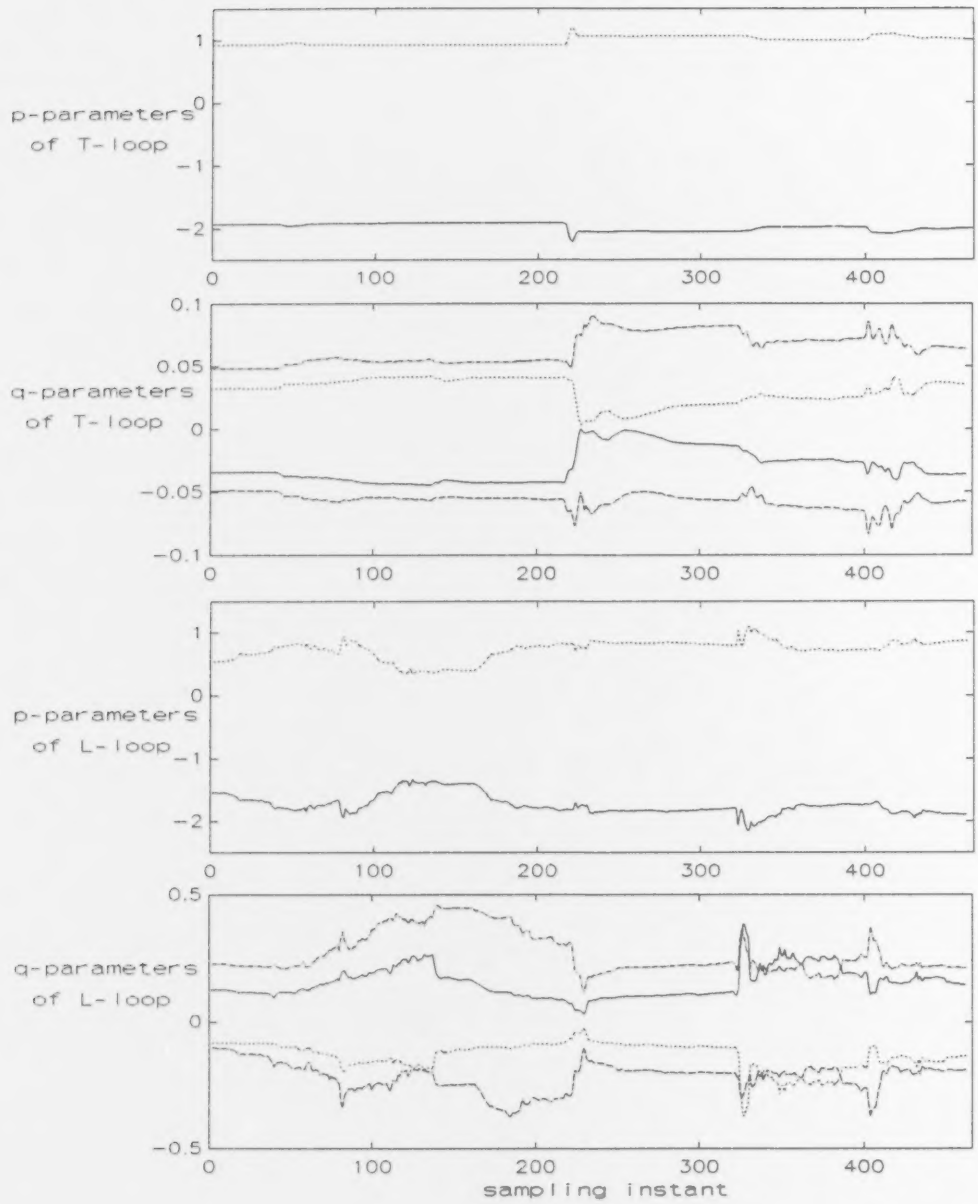


Figure 5.25 Parameter trajectories of the MISO subsystems corresponding to the temperature and level loops in ARMAX representation in Constrained MAdGPC.

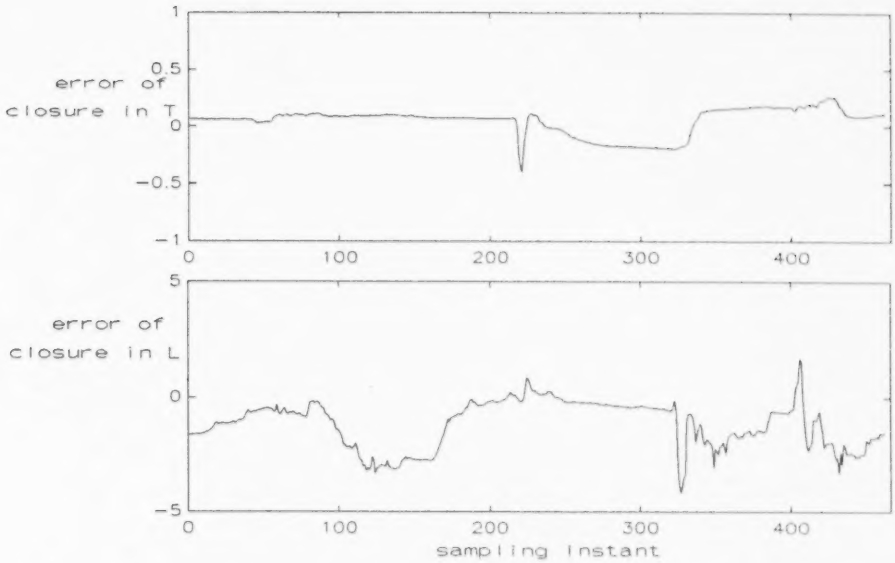


Figure 5.26 Estimates of the error of closure in temperature and level in ARMAX representation in Constrained MAdGPC.

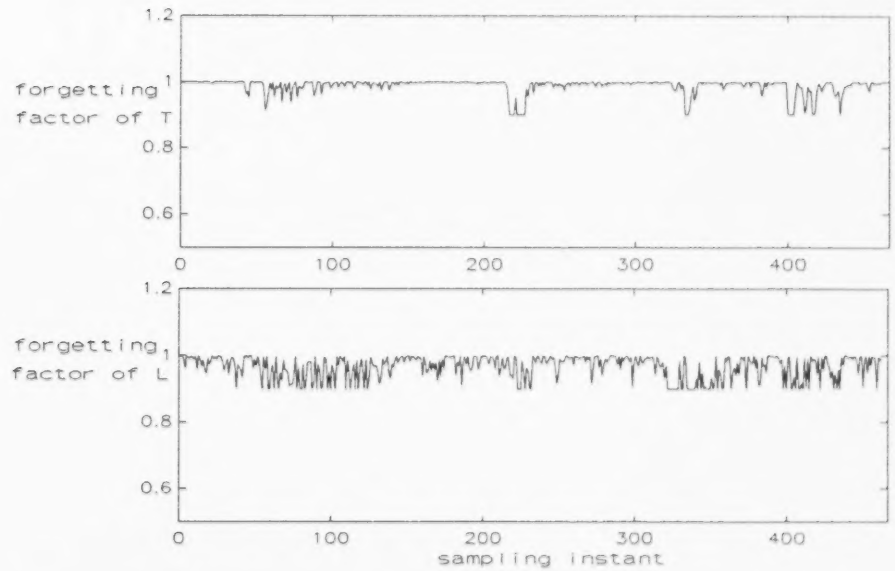


Figure 5.27 Variation of forgetting factor in Constrained MAdGPC.

very effectively. An "impulse type" disturbance was introduced in the level loop by suddenly shuting the exit valve and then re-opening it at sampling instant 330. A sudden increase in the level is the result of this disturbance but the controller is able to bring back the level to its desired set-point value.

5.2.3.11 Multi-loop PID controller implemented on a 2 x 2 Process

This experimental run was conducted to study the performance of multi-loop PID controllers performance on the same process, for comparison with MAdGPC. The pairing of outputs to inputs was done based on the relative gain arrays (please refer to Mutha, 1991). The temperature was controlled by manipulating cold water valve and the level in the tank was paired to hot water manipulating valve.

The tuning of the multiloop PID controllers was a laborious task. The tuning constants of the PID for temperature loop were $K_p = -0.07$, $K_i = -0.03$, $K_d = -0.14$ and for the level loop were $K_p = 0.5$, $K_i = 0.02$, $K_d = 0.5$. The performance of multiloop PID controllers is shown in figure 5.28. It can be seen that compared to MAdGPC the performance of multiloop PID is poor. There are large interactions between the two loops. Although, low integral gains and high derivative gains of the controller were used, there are still large oscillations in the outputs of this process. The disturbance effects were not investigated since the normal response was so oscillatory. Thus, it can be concluded that multiloop PID is far far inferior to MAdGPC when controlling a multivariable process.

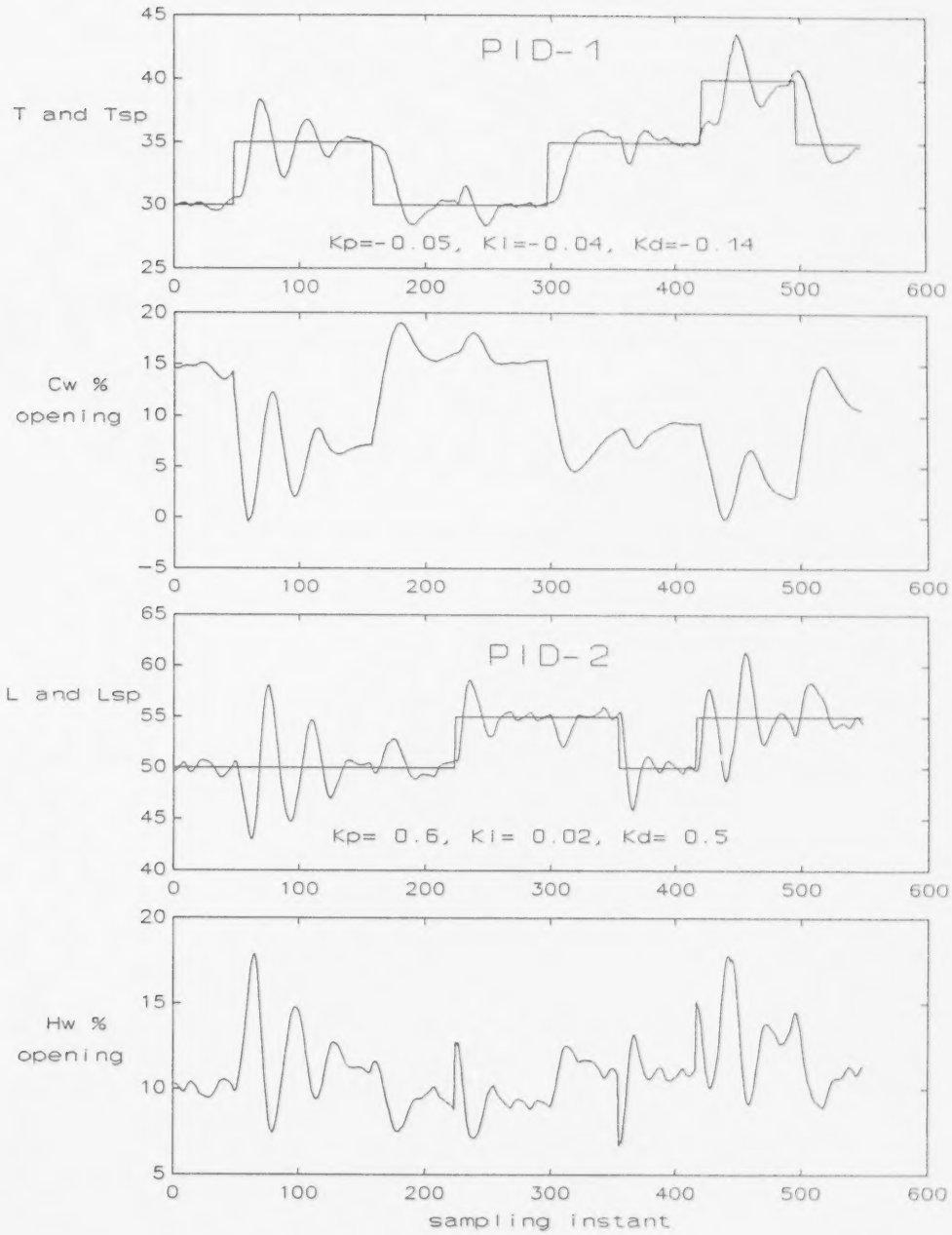


Figure 5.28 Multi-loop PID implemented on the continuous stirred tank heater configured as a 2 x 2 process.

In this chapter, the MAdGPC algorithm has been evaluated by conducting several experimental runs. The importance of each of its tuning parameters and the resulting guidelines on how to choose their values has been provided in a form of a specific case study. The algorithm has been tested with and without the presence of constraints. Superior performance of this adaptive algorithm in comparison to a classical multiloop PID controller has also been established.

Chapter 6

Conclusions

This thesis has successfully extended the augmented UD (AUD) identification algorithm to multi-input multi-output (MIMO) systems with distinct time-delays. The MIMO AUD identification algorithm (MIMO AUDI) was realized by using the idea of decomposing a MIMO system into a number of multi-input single output (MISO) subsystems. It was evaluated by using the input-output data from a real process. This algorithm can simultaneously identify the model order of each of the MISO subsystems and the model parameters on-line.

A multi-input multi-output (MIMO), adaptive long-range predictive controller (LRPC) has been successfully evaluated on a pilot-scale process. The MIMO adaptive LRPC was implemented by using the new MIMO AUD identification algorithm in conjunction with the MIMO Generalized Predictive Control (MGPC) algorithm. The performance of this adaptive controller was evaluated on a pilot-scale continuous stirred tank heater in the presence of unmeasurable disturbance effects, plus constraints on the absolute and incremental inputs and outputs of the process.

The main contributions of this thesis are:

1 Transformation of a MIMO system into MISO subsystems for identification

A multivariable system can be represented generally in the P-canonical form or the transfer function matrix form. It is not convenient to use any of the popular recursive identification techniques to identify systems with these structures. The most convenient form for representing a system for identification is in the auto-regressive moving average (ARMA) form. Transformation of a single-input, single-output (SISO) system from the transfer function form to the ARMA form is straight forward, but this is not the case for multivariable systems. This thesis provides the transformation from a multivariable system in transfer function matrix form to the ARMA or input-output difference equation form via a state-space representation. The transformed multivariable system in ARMA form always has its output polynomial matrix, $P(z^{-1})$ in diagonal form and the input polynomial matrix, $Q(z^{-1})$, in full matrix form. This special multivariable ARMA form can be subsequently decomposed into a set of MISO subsystems that are very convenient for identification.

2 MIMO AUD identification algorithm

The AUD identification algorithm has been successfully extended to identify multivariable systems with distinct time-delays. This extension is based on the ARMA structure representation of the MIMO system. Because of the special form of MIMO ARMA structure, the MIMO system can be decomposed into as many MISO subsystems, as the number of outputs of the system. This strategy of decomposing MIMO system into

MISO subsystems is found to be convenient for handling distinct time delays in the original MIMO system. The MIMO AUDI algorithm can simultaneously identify model order and process parameters on-line. The performance of this algorithm was evaluated on data collected from a real process. A variable forgetting factor was incorporated in this algorithm and shown to enhance the adaptivity of the estimation scheme. It was shown that a lower bound on the value of forgetting factor is useful for reducing variance in the parameter estimates.

3 Evaluation of MIMO adaptive GPC

The performance of an MIMO adaptive GPC algorithm was evaluated experimentally on a pilot-scale continuous stirred tank heater. The effects of various parameters on the performance of the algorithm were studied.

It was shown that the value of the sigma parameter in calculating the forgetting factor value in the identification part of the adaptive controller is important and if improperly used may lead to an unstable controller. The lower bound on the value of forgetting factor also plays an important role in the performance of the controller indirectly, since it governs the variance of the parameter estimates of the process.

The effect of the MIMO adaptive GPC controller tuning parameters such as the maximum prediction horizon (N_2), input weighting (Q), output weighting (y_wt) were also investigated by experimental evaluation. It is shown that a proper value of N_2 is very important, for almost open-loop unstable process such as the one discussed in this thesis. For large values of N_2 the controller behaves as a mean-level controller with closed-loop

poles close to the open-loop positions. Input weighting is also a very important tuning parameter especially when there is "ringing" occurring in the process due to large time constants of the process. The "ringing" behaviour can be considerably reduced by increasing the weights on the inputs, although this generally gives a sluggish response. Output weighting is also a crucial tuning parameter, since it dictates or allows one to decouple the output from the effect of setpoint changes in the other outputs by giving it a larger weight compared to the other outputs. It is also possible to achieve "symmetric or balanced control" performance by scaling the individual loops of MAdGPC such that under typical operating conditions the values of the error used in the control calculations after analog to digital conversion and normalization are of similar magnitude.

The performance of the MIMO adaptive controller was also investigated in the presence of rate and amplitude constraints on the inputs and amplitude constraints on the outputs. It was shown that it is possible to get stable control performance for a poorly tuned controller by using tight rate constraints on the inputs. Also the "ringing" can be considerably reduced when rate constraints are used with higher input weightings. Amplitude constraints on the input action are mainly used to force the controller to give physically realizable signals. It was shown that output constraints were operational and that the GPC controller is capable of maintaining the outputs within certain limits under the effect of external disturbances. The analytical solutions of the constrained MAdGPC controller were also successfully evaluated by application to a real process.

The MIMO adaptive GPC controller provides superior setpoint tracking properties, is able to reject disturbances very well and can maintain the output at the

desired value with and without constraints on the rate and amplitude of the inputs and on the amplitude of the outputs. The performance of a multiloop PID was shown to be inferior in controlling the same 2 x 2 plant compared to the performance of unconstrained MIMO adaptive GPC.

Recommendations for Future Work

It was shown that the MIMO AUD algorithm is capable of identifying the model order and the parameters on-line, but it needs fairly precise information about the time delays in each element of the MIMO process. If the time-delay of the process is identified on-line then it would give better estimates of the parameters.

The MIMO adaptive GPC algorithm has been shown to be a superior algorithm, as it can control open-loop unstable, highly interacting, non-linear plants, plants with variable time-delays. This algorithm has been evaluated on a pilot-scale continuous stirred tank heater. The robustness aspects of this algorithm should be studied to make it more robust and to control a highly non-linear, time varying processes like polymer reactor or a bio-reactor. Such processes can be good candidates for implementing a superior algorithm like MIMO adaptive GPC.

References

1. Åström K. J and B. Wittenmark, 1973. "On self-Tuning Regulators". Automatica, 9 (2), 185-199.
2. Åström K. J and B. Wittenmark, 1989. **Adaptive Control.** Addison-Wesley Co.
3. Beghelli S. and R. P. Guidorzi, 1983. " Transformation between input and output multistructural models: Properties and applications". Int. J. of Control, 37 (4), 1385-1400.
4. Bierman G. J, 1977. **Factorization Methods for Discrete Sequential Estimation.** Academic Press, New York.
5. Clarke D. W and P. J. Gawthrop, 1975. "Self-Tuning Controller". Proc. IEE 122 (9), 929-934.
6. Clarke D. W, 1981. Implementation of self-tuning controllers. Self-Tuning and Adaptive Control: Theory and Applications. Edited by C. S. Harris and S. A. Billings (London, Peter Peregrinus).
7. Clarke D. W and C. Mohtadi, 1989. "Properties of Generalized Predictive Control". Automatica, 25 (6), 859-875.

8. Clarke D. W, C. Mohtadi and P. S. Tuffs, 1987. "Generalized Predictive Control-Part I. The Basic Algorithm". Automatica, 23 (2), 137-148.

9. Clarke D. W, C. Mohtadi and P. S. Tuffs, 1987. "Generalized Predictive Control-Part II. Extensions and Interpretations". Automatica, 23 (2) 149-160.

10. Cutler C. R and B. L. Ramaker, 1980. "Dynamic Matrix control - A computer control algorithm". Proc. Joint Automatic Control Conference, San Francisco, U.S.A.

11. De Keyser R. M. C and A. R. Van Cauwenberghe, 1985. "Extended Prediction Self-Adaptive Control". Proc. of the 7th IFAC Symposium on Identification and System Parameter Estimation, York, U.K.

12. Fortescue T. R, L. S. Kershenbaum and B. E. Ydstie, 1981. "Implementation of Self-Tuning Regulators with Variable Forgetting Factors". Automatica, 17 (6) 831-835.

13. Gill P. E, W. Murray, M. A. Saunders and M. H. Wright, 1984. "Users' Guide for QPSOL: A Fortran Package for Quadratic Programming". Technical Report SOL 84-6, Dept. of Operations Research, Stanford University, Stanford, U.S.A.

14. Goodwin G. C and K. S. Sin, 1984. **Adaptive Filtering, Prediction and Control.** Prentice-Hall, Englewood Cliffs, New Jersey, U.S.A.

15. Goodwin G. C, H. Elliot and E.K Teoh, 1983. "Deterministic convergence of a self-tuning regulator with covariance resetting". Proc. of IEE, 130 part D, 6-8.

16. Goodwin G. C, D. J. Hill and M. Palaniswami, 1985. "Towards an adaptive robust controller". Proc. of the IFAC conference on Identification and System

Parameter estimation, York, U.K.

17. Guidorzi R. P, 1975. "Canonical Structure in the identification of multivariable systems". Automatica, 11, 361-374.

18. Guidorzi R. P, 1981. "Invariant and Canonical forms for system, structural and parametric identification". Automatica, 17 (1), 117-133.

19. Hostetter G. H, 1988. **Digital Control System Design.** Holt, Rhinehart and Winston, Inc. The Dryden Press, Saunders College Publishing.

20. Isermann R., 1981. **Digital Control Systems.** Springer-Verlag Berlin, Heidelberg, New York, U.S.A.

21. Kailath T., 1980. **Linear systems.** Prentice-Hall, Englewood Cliffs, New Jersey, U.S.A.

22. Lee W. and V. W. Weekman. Jr., 1976. "Advanced Control Practice in Chemical Process Industry: A view from Industry". AIChE Journal, 22, 27.

23. Ljung L., 1987. **System Identification: Theory for the User.** Prentice-Hall, Englewood Cliffs, New Jersey, U.S.A.

24. Ljung L. and T. Söderström, 1983. **Theory and Practice of Recursive Identification.** MIT Press, Cambridge, Mass U.S.A.

25. Ljung S. and L. Ljung, 1985. "Error propagating properties of recursive least-squares adaptive algorithms". Automatica, 21, 157-167.

26. Luyben W. L, 1989. **Process Modelling, simulation and Control for Chemical Engineers.** Mc Graw Hill.

27. Maurath P. R, D. E. Seborg and D. A. Mellichamp, 1986. "Achieving

Decoupling with Predictive Controllers". Proc. ACC, Seatle, 3, 1372- 1377.

28. McIntosh A. R, 1988. Performance and Tuning of Adaptive Generalized Predictive Control. M.Sc. Thesis, Dept. of Chemical Engineering, University of Alberta.

29. McIntosh a. R, S. L. Shah and D. G. Fisher, 1991. "Analysis and Tuning of Adaptive Generalized Predictive Control". CJChE, 69, 97-110.

30. Mohtadi C., 1987. Advanced Self-Tuning Algorithms. D. Phil Thesis, University of Oxford.

31. Mohtadi C., S. L. Shah and D. W. Clarke, 1986. "Generalized Predictive Control of Multivariable Systems". Report No. OUEL 1640/86.

32. Mutha R. K, 1990. Constrained Long Range Predictive Control. M.Sc. Thesis, Dept. of Chemical Engineering, University of Alberta.

33. Niu S., D. Xiao and D. G. Fisher, 1990. "A Recursive algorithm for simultaneous identification of model order and parameters". IEEE Trans. on Acoustics, Speech and Signal Processing, 38 (5), 884-886.

34. Niu S. and D. G. Fisher, 1991. "MIMO System Identification using Augmented UD Factorization". ACC, Boston, Mass U.S.A.

35. Ortega R., L. Praly and I.D. Landau, 1985. "Robustness of discrete-time adaptive controllers". IEEE Trans. of Automatic Control, 39, 1179-1187.

36. Peterka V., 1984. "Predictive based self-tuning control". Automatica, 20 (1), 39-50.

37. Powell M. J. D, 1989. "TOLMIN: A Fortran Package for Linearly Constrained

Optimization Calculations". Numerical Analysis Reports DAMTP 1989/NA2, Dept. of Applied Mathematics and Theoretical Physics, University of Cambridge, Cambridge.

39. Praly L., 1983. "Robustness of Model Reference Adaptive Control". Proc. of 3rd Yale Workshop on Adaptive Control, New Haven, CT, U.S.A.

40. Qiu, 1989. MULTICON users manual, University of Alberta.

41. QNX Reference Guide, Version 2.1, Quantum Software Systems Ltd., Kanata, Ontario, Canada.

42. Richalet J., A. Rault, J. L. Testud and J. Papon, 1978. "Model Predictive Heuristic Control: Applications to Industrial Processes". Automatica, 14 (5), 413-428.

43. Sastry S. and M. Bodson, 1989. **Adaptive Control: Stability, Convergence and Robustness.** Prentice-Hall, Englewood Cliffs, N.J.

44. Seborg D. E, T F. Edgar and S. L. Shah, 1986. "Adaptive Control Strategies for Process Control: A survey". AIChE Journal, 32 (6), 881-913.

45. Shah S. L and W. R. Cluett, 1991. "Recursive Least Squares based estimation schemes". CJChE, 69, 89-96.

46. Shah S. L, C. Mohtadi and D. W. Clarke, 1987. "Multivariable adaptive control without a prior knowledge of the delay matrix". Systems and Control Letters, 9, 295-306.

47. Thornton C. L and G. J. Bierman, 1980. "UDU^T covariance factorization for kalman filtering". Control and Dynamic Systems, 16, .

48. Tsang T. T. C and D. W. Clarke, 1988. "Generalized Predictive Control with Input constraints". IEE part D, 135 (6), 451-460.

49. Wolovich and Elliot, 1983. "Discrete models for Linear Multivariable Systems". Int. J. Control, 38 (2), 337-357.

50. Ydstie B. E, 1984. "Extended Horizon Adaptive Control", Proc. of 9th IFAC World Congress, Budapest, Hungary.